Cranfield University

Anna Malahova

A Methodology for Developing Web-based CAD/CAM Systems: Case Studies on Gear Shaper Cutters

School of Applied Sciences

PhD Thesis

Cranfield University

School of Applied Sciences

PhD Thesis

Academic Year 2013-2014

Anna Malahova

A Methodology for Developing Web-based CAD/CAM Systems: Case Studies on Gear Shaper Cutters

Supervisor: Prof. A. Tiwari

June 2014

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Contents

Contents i

Abstract vii

Acknowledgements ix

List of figures xi

List of tables xv

List of publications xvii

List of acronyms xix

1 Introduction **1**

1.1 Introduction to software development 1
1.2 Web-based applications 3
1.3 CAD/CAM and CSE software 3
1.4 Sponsoring company and KTP project 6
1.5 Problem statement and motivation 7
1.6 Thesis content 8
1.7 Summary 9

2 Literature Review **11**

2.1 Web technology in industry 11
2.1.1 Web technology overview 12
2.1.2 Application areas 13
2.2 Motivation for Web 15
2.2.1 Software development trends 16
2.2.2 Advantages of Web-based approach 16

2.3 Web-based approach in product design 17
2.3.1 Classification of Web-based CAD/CAM systems 18
2.3.2 Web-based CAD/CAM implementation methods and technologies 19
2.3.3 Challenges of Web-based CAD/CAM development 22
2.3.3.1 Internet connectivity and security 22
2.3.3.2 Interactivity and graphics visualisation 23
2.3.3.3 Sharing tasks between the server and the client 23
2.3.3.4 User collaboration 24
2.3.3.5 Design and development 24
2.4 Software development methodologies 25
2.4.1 Evolution of software development methodologies 25
2.4.2 Methodologies relevant to the research 28
2.4.2.1 Web application development 28
2.4.2.2 CAD/CAM software development 29
2.4.2.3 Scientific CAD/CAE and complex CSE software development 29
2.4.3 Application of software development methodologies 30
2.5 Research gap 35
2.6 Summary 35

3 Research aim, objectives and methodology 39
3.1 Research aim and objectives 39
3.2 Research methodology 40
3.2.1 Analysis 40
3.2.2 Development 43
3.2.3 Application 46
3.2.4 Validation and evaluation 48
3.3 Summary 49

4 Rationale and prerequisites for methodology 51
4.1 Science-intensive software 51
4.2 Requirements elicitation 52
4.3 Design 53
4.4 Planning 54
4.5 Incremental development challenges 55
4.6 Features specific to Web-based CAD/CAM software 57

4.7 Web-based CAD/CAM software development challenges 58
4.8 Prerequisites for specialised methodology 58
4.9 Summary 60

5 A methodology for Web-based CAD/CAM systems 63
5.1 Major considerations and key principles 63
5.2 Software development process 66
5.2.1 Project phases 66
5.2.2 Development process model 66
5.3 Development planning 68
5.4 Software design optimisation 72
5.4.1 Web-based CAD/CAM software product concept 72
5.4.2 Critical design decision points 75
5.5 Addressing Web-based CAD/CAM development concerns 77
5.5.1 Performance 78
5.5.2 Robustness 78
5.5.3 Compatibility and interoperability 79
5.5.4 Flexibility 80
5.5.5 Scalability 81
5.5.6 Reliability 83
5.5.7 Security 83
5.5.8 Usability 84
5.5.9 Maintainability 85
5.6 Requirements elicitation 86
5.7 Overcoming uncertainty 87
5.8 Resource allocation and collaboration 89
5.9 Applicability 90
5.10 Projected benefits from applying the methodology 92
5.11 Summary 92

6 CAD/CAM system for gear shaper cutters 95
6.1 Design and manufacture of gear shaper cutters 95
6.2 Case study description 96
6.2.1 Software concept and purpose 97
6.2.2 Business processes and workflow 98
6.3 Eliciting user requirements 100

6.4 Planning the development 103
6.5 Resource allocation and collaboration 107
6.6 Creating optimal design of the software 108
6.7 Software architecture and implemented technologies 113
6.8 User interface 116
6.9 Challenges and solutions 134
6.9.1 Evolution of the applied analytical model of gearing 135
6.9.2 Extensive computations, debugging and maintenance 138
6.9.3 User interface development 139
6.9.4 Extending user requirements 139
6.9.5 Tooth profile error problem 140
6.10 Impact of the development of Web-based CAD/CAM for GSCs 141
6.11 Summary 142

7 Web-based CNC code editor 145
7.1 Case study description 145
7.1.1 Software concept and purpose 146
7.1.2 Business processes and workflow 148
7.2 Eliciting user requirements 150
7.3 Planning the development 150
7.4 Resource allocation and collaboration 154
7.5 Creating optimal design of the software 155
7.6 Software architecture and implemented technologies 159
7.7 User interface 162
7.8 Challenges and solutions 163
7.8.1 Ellipse approximation for the CNC code 163
7.8.2 Tracking geometry changes in large CNC programs 163
7.9 Impact of the development of Web-based CNC code editor 164
7.10 Summary 165

8 Validation 167
8.1 CAD/CAM system for spur involute gear shaper cutters 167
8.1.1 Real design examples 167
8.1.2 System evaluation through industrial use 176
8.1.3 Industrial feedback 179
8.1.4 Analysis of advantages and limitations 181

8.2 Web-based CNC code editor . . . 183
8.2.1 Usage example . . . 183
8.2.2 System evaluation through industrial use . . . 185
8.2.3 Industrial feedback . . . 185
8.2.4 Analysis of advantages and limitations . . . 186
8.3 Summary . . . 188

9 Discussion and conclusions 191
9.1 Key findings . . . 191
9.1.1 Previous research . . . 191
9.1.2 Rationale and prerequisites for methodology . . . 192
9.1.3 A methodology for creating Web-based CAD/CAM systems . . . 193
9.1.4 Application of the methodology . . . 194
9.1.5 Validation of the methodology . . . 195
9.2 Discussion . . . 196
9.2.1 Major contributions . . . 197
9.2.2 Limitations . . . 198
9.2.3 Future research . . . 198
9.3 Conclusions . . . 199

References 202

Appendices 211

A Design case 1 211
A.1 Web-based CAD/CAM system outputs . . . 211
A.1.1 Tool drawing . . . 211
A.1.2 Design details . . . 213
A.1.3 Multigraph . . . 216
A.1.4 Job ticket . . . 218
A.1.5 Former . . . 220
A.1.6 Haas CNC instructions . . . 222

B Design case 2 225
B.1 Design data . . . 225
B.2 Web-based CAD/CAM system outputs for design case 2 . . . 226
B.2.1 Tool drawing . . . 226

B.2.2 Design details 228
B.2.3 Multigraph 230
B.2.4 Job ticket 232
B.2.5 Former 234
B.2.6 Haas CNC instructions 236

C Design case 3 **237**
C.1 Design data 237
C.2 Web-based CAD/CAM system outputs for design case 3 239
C.2.1 Tool drawing 239
C.2.2 Design details 241
C.2.3 Multigraph 244
C.2.4 Job ticket 246
C.2.5 Former 248
C.2.6 Diaform CNC instructions 250

D CNC program 1 for internal master gear design **251**
D.1 Original CNC program 1 251
D.2 Modified CNC program 1 252

E CNC program 2 for external master gear design **255**
E.1 Original CNC program 2 255
E.2 Modified CNC program 2 256

F CNC program 3 for standard cutter design **257**
F.1 Original CNC program 3 257
F.2 Modified CNC program 3 258

G Questionnaire: the Web-based CAD/CAM system for GSCs **259**

H Questionnaire: the Web-based CNC code editor **267**

Abstract

The research establishes a methodology for developing Web-based CAD/CAM software systems to industrial quality standards in a time and cost effective manner. The methodology defines the scope of applicability, outlines major considerations and key principles to follow when developing this kind of software, describes an approach to requirements elicitation, resource allocation and collaboration, establishes strategies for overcoming uncertainty and describes the design concerns for industrial Web-based CAD/CAM systems. The crucial parts of the methodology are a novel project development model facilitating architecture optimisation early in the project to minimise total development efforts, create future-proof solutions and ensure system maintainability; and a novel approach for planning based on time reserve management and task prioritisation, which provides the flexibility required for exploratory development while maintaining the main focus on project objectives.

The effectiveness of the Web-based CAD/CAM software development methodology has been examined using two real software development case studies: a Web-based CAD/CAM system for involute spur gear shaper cutters and a Web-based CNC code editor for online modification of the profile for manufacturing gear shaper cutters. The development of case studies using the established methodology resulted in on-time delivery of two industrial browser-based CAD/CAM systems, that produce valid results, embrace all business processes associated with the application area, ensure all functional and non-functional requirements and are used in production now. The developed software products demonstrate robustness, performance, reliability, security and usability comparable with the standards of modern commercial software, utilise advantages of Web-based applications to the highest extent and confirm advantages of Web-based CAD/CAM software compared to similar desktop applications. Effectiveness of the proposed methodology for Web-based CAD/CAM software development was checked through validation, evaluation and analysis of case study results.

Acknowledgements

This work was supported by a Knowledge Transfer Partnership awarded to Dathan Tool & Gauge Co. Ltd. and the Cranfield University. I would like to take this time to thank the parties for all of the funding and support they were able to provide to me in order to make this thesis possible.

I would also like to thank Mr. Richard Moorhouse, Mr. Chris Moorhouse and my colleagues from Dathan Tool & Gauge Co. Ltd. for sharing their immense knowledge and expertise with me and for their contribution to the application and validation part of this research.

I would like to express my sincere gratitude to my supervisor Prof. Ashutosh Tiwari for the continuous support of my Ph.D study, for his patience, motivation, knowledge and guidance.

I would like to thank the rest of my thesis committee for their encouragement, insightful comments, and hard questions.

I would also like to thank Dr. Yuchun Xu and Mr. Tim Brewer as they both provided me with many great points to include and gave me advice when it was required.

I would also like to thank my friend Dr. Eugene Butan for his support, encouragement and advice, I appreciate that very much.

List of Figures

1.1 Multidisciplinary nature of CSE 5
1.2 CAD/CAM incorporates industrial application and CSE features 5
1.3 Gear cutting tools supplied by Dathan 6

2.1 W3C activity creation timeline from 1994 up to 2011 13
2.2 Number of publications about Web-based systems 14
2.3 Number of publications about web-based PDM 14
2.4 Number of publications about Web-based decision support systems . . . 15
2.5 Subject areas of Web-based DSS research 15
2.6 Percentage of separate Web-based CAD/CAM solutions versus integrated with proprietary CAD/CAM software 20
2.7 Prototype software versus industrial Web-based CAD/CAM applications 20
2.8 Percentage of Web-based CAD systems implementing Java technology . . 21
2.9 Visualisation technologies used in Web-based CAD systems 21
2.10 Evolution of software development methodologies 26

3.1 Conceptual diagram of the research methodology 40
3.2 Steps, methods and tools used in the analysis stage 41
3.3 Flow chart of the research process in the analysis stage 42
3.4 Steps, methods and tools used in the development stage 43
3.5 Providing rationale and prerequisites for Web-based CAD/CAM development methodology 43
3.6 Steps and decisions used to construct a methodology for Web-based CAD/CAM development 44
3.7 Concerns for Web-based CAD/CAM software development methodology 45
3.8 Steps, methods and tools used in the application stage 46
3.9 Flowchart of the research process in the application stage 47
3.10 Steps, methods and tools used in the validation and evaluation stage . . 48

3.11 Method for validating the proposed methodology for Web-based CAD/CAM development 49

4.1 Balancing design and refactoring 53
4.2 Software build-up in incremental development 55
4.3 Moment of inertia at different distances between the current state and change 57
4.4 Deriving prerequisites for the specialised methodology from features specific to Web-based CAD/CAM software and development challenges . . . 59

5.1 Project development process model 68
5.2 Planning task composition for each iteration depending on the intensity of using reserved effort 69
5.3 A block scheme showing the process of decision making about the task composition for each iteration 71
5.4 Web-based CAD/CAM software product considerations 73
5.5 Web-based CAD/CAM system interactivity 76
5.6 Task distribution between the server and the client 76
5.7 Software collaboration concerns 77
5.8 Addressing software performance concerns 78
5.9 Addressing software robustness concerns 79
5.10 Key considerations for software compatibility 79
5.11 Client side compatibility 80
5.12 Addressing software flexibility concerns 81
5.13 Distributed systems scalability aspects 82
5.14 Addressing software reliability concerns 83
5.15 Addressing software security concerns 84
5.16 Key considerations for software usability 85
5.17 Addressing software maintainability concerns 86

6.1 Mind-map containing different considerations related to the Web-based CAD/CAM system for GSCs 99
6.2 Gear shaper cutter design process 101
6.3 Process model for developing the Web-based CAD/CAM system for GSCs 103
6.4 Work breakdown for the development of the Web-based CAD/CAM system for GSCs 104

6.5 Interactivity related design decisions for the Web-based CAD/CAM system for GSCs . . . 108
6.6 Collaboration related design decisions for the Web-based CAD/CAM system for GSCs . . . 109
6.7 Addressing design considerations for the Web-based CAD/CAM system for GSCs using prototypes for user interface, computational engine, analytical model of gearing and visualisation and interactivity module . . . 110
6.8 Entity-relationship diagram of Web-based CAD/CAM system for gear shaper cutters . . . 111
6.9 Use case diagram of Web-based CAD/CAM system for gear shaper cutters 112
6.10 Architecture of Web-based CAD/CAM system for gear shaper cutters . . 114
6.11 User authentication and profile data editing . . . 117
6.12 System main screen . . . 119
6.13 Multigraph and tooth profile inspection tools . . . 121
6.14 Visualisation of the roll motion when generating gear with the cutter . . 122
6.15 Interference prediction between internal gear and cutter profiles . . . 123
6.16 Dependency of gear chamfer and root diameter on tool life . . . 123
6.17 Manufacturing tool selection . . . 124
6.18 Cutter body shape design . . . 126
6.19 Tool face drawing . . . 127
6.20 Additional design information . . . 128
6.21 Open design dialogue window . . . 128
6.22 Save design dialogue window . . . 129
6.23 Edit gear list dialogue window . . . 130
6.24 Gear circular thickness calculator . . . 130
6.25 Dialogue to edit manufacturing tool data . . . 131
6.26 User account administration . . . 132
6.27 User group membership management . . . 132
6.28 Exporting drawings in PDF . . . 133
6.29 Design import from file . . . 134
6.30 CNC code generation . . . 135
6.31 Conjugate profiles with and without gear profile error on the undercut segment . . . 141

7.1 Generation of a helical cutter with a grinding wheel . . . 146

7.2 Mind-map containing different considerations related to the Web-based CNC code editor . . . 147
7.3 CNC code editing process . . . 149
7.4 Process model for developing the Web-based CAD/CAM system for GSCs 151
7.5 Work breakdown for the Web-based CNC code editor development 152
7.6 Addressing design considerations for the Web-based CNC code editor using prototypes for user interface, profile approximation procedure and module for tracking geometry changes . . . 156
7.7 Entity-relationship diagram of Web-based CNC code editor for gear shaper cutters . . . 158
7.8 Use case diagram of Web-based CNC code editor for gear shaper cutters 159
7.9 Architecture of the Web-based CNC code editor for gear shaper cutters . 160
7.10 CNC code editor . . . 162

8.1 Design master gear tooth profile . . . 169
8.2 Cutter tooth profile design from master gear data and redesign at different pressure angle . . . 170
8.3 Involute deviation graph for master gear design and redesign 170
8.4 Involute deviation graph for the secondary gears with minimum and maximum allowed circular tooth thickness . . . 171
8.5 Variations in the chamfer and root diameters of the design master and secondary gears throughout the cutter life . . . 173
8.6 Manufacturing visualisation . . . 175
8.7 Tool body drawing . . . 176
8.8 Evaluation of the Web-based CAD/CAM system compared to the desktop CAD/CAM system based on questionnaire results . . . 180
8.9 Drawing of the profile and selection of coordinates for tool path correction 184
8.10 Evaluation of the Web-based CNC code editor compared to the desktop software based on questionnaire results . . . 186

B.1 Rack tooth profile . . . 225

C.1 Cutter and internal gear teeth profile interference analysis. 238

List of Tables

2.1 Comparison of software development models 31
2.2 Positive and negative software engineering experiences 34

5.1 Major considerations for Web-based CAD/CAM software development . 64
5.2 Task priority evolution throughout the project development 70

6.1 Prioritising development tasks for the Web-based CAD/CAM system for GSCs 105
6.2 Time reserve management for the development of the Web-based CAD/CAM system for GSCs 106
6.3 Engagement of different stakeholders in Web-based CAD/CAM development 107
6.4 Design decisions about task distribution between the server and the client in the Web-based CAD/CAM system for GSCs 109
6.5 Deployment requirements for the Web-based CAD/CAM system for GSCs 115

7.1 Prioritising development tasks for the Web-based CNC code editor 153
7.2 Time reserve management for the development of the Web-based CNC code editor 154
7.3 Engagement of different stakeholders in Web-based CNC code editor . . . 155
7.4 Task distribution between the server and the client in the Web-based CNC code editor 157
7.5 Deployment requirements 161

8.1 Cutter details 168
8.2 External master gear details 169
8.3 Multiple gears 171
8.4 Design results 174
8.5 Manufacturing details 175
8.6 Tool body details 177

8.7 Real designs produced using the Web-based CAD/CAM system 178
8.8 Resolved issues discovered in the Web-based CAD/CAM system for GSCs 179
8.9 Priority of issues in the Web-based CAD/CAM system for GSCs 179
8.10 Advantages and limitations of the Web-based CAD/CAM system for GSCs 182
8.11 Set of coordinates for the example profile 184
8.12 CNC programs modified using the Web-based CNC code editor 185
8.13 Advantages and limitations of the Web-based CNC code editor 187

B.1 Cutter details 225
B.2 Tool body details 226

C.1 Internal master gear details 237
C.2 Cutter details 238
C.3 Tool body details 239

List of publications

1. A. Malahova, J. Butans, A. Tiwari. A Web-based CAD System for Gear Shaper Cutters. The Cranfield Multi-Strand Conference: Creating Wealth Through Research and Innovation, 6 - 7 May 2008, Cranfield, United Kingdom, pp. 527-532.

2. A. Malahova, A. Tiwari. IT for Product Design. KTP Associates Conference, 29 April 2009, Brighton, United Kingdom, pp. 63-68.

3. A. Malahova, J. Butans, A. Tiwari. A Web-based CAD System for Gear Shaper Cutters. The 7th IEEE International Conference on Industrial Informatics, INDIN 2009, Cardiff, United Kingdom, pp. 37-42.

4. A. Malahova, A. Tiwari. Methodology for developing Web-based CAD/CAM software. International Journal of Science and Engineering Applications 3 (6) (2014)

List of acronyms

AI Artificial Intelligence

AJAX Asynchronous JavScript and XML

API Application Programming Interface

ASP Active Server Pages

CAD Computer-Aided Design

CAD/CAM Computer-Aided Design and Manufacturing

CAE Computer-Aided Engineering

CAM Computer-Aided Manufacturing

CCPM Critical Chain Project Management

CMM Coordinate Measurement Machine

CNC Computer Numerical Control

CPU Central Processing Unit

CRM Customer Relationship Management

CSE Computational Science and Engineering

DB Database

DSDM Dynamic Systems Development Method

DSS Decision Support System

ERM Entity-Relationship Model

ERP Enterprise Resource Planning

FDD Feature Driven Development

GSC gear shaper cutter

HTML Hyper Text Markup Language

HTML5 The fifth revision of the Hyper Text Markup Language standard

IDE Integrated Development Environments

JRE Java Run-time Environment

JSON JavaScript Object Notation

KTP Knowledge Transfer Partnership

LOC Lines Of Code

MDA Model Driven Architecture

MVC Model-View-Controller

OOP Object-Oriented Programming

PDF Portable Document Format

PDM Product Data Management

PIL Python Imaging Library

PLM Product Lifecycle Management

PRA Probabilistic Risk Assessment

RAD Rapid Application Development

RAID Redundant Array of Independent Disks

RAM Random Access Memory

RM-ODP Reference Model for Open Distributed Processing

RSDM Robust Software Development Model

SDM System Development Methodology

SDM2 Cap Gemini System Development Methodology

SISOS Software-Intensive Systems of Systems

SME Situational Method Engineering

SQL Structured Query Language

SSADM Structured Systems Analysis and Design Methodology

SSL Secure Sockets Layer

SSM Soft Systems Methodology

SVG Scalable Vector Graphics

TDD Test-Driven Development

UI User Interface

VPN Virtual Private Network

VRML Virtual Reality Modelling Language

WebGL Web Graphics Library

WIS Web Information Systems

W3C World Wide Web Consortium

WWW World Wide Web

XML Extensible Markup Language

XP Extreme Programming

Chapter 1

Introduction

Developing a Web-based Computer-Aided Design and Manufacturing (CAD/CAM) system involves expertise in several fields of knowledge: software engineering, Web-based systems, CAD/CAM and the specialist domain expertise. The purpose of this chapter is the understanding of underlying terms and principles in the respective areas of knowledge.

The subsequent sections in this chapter introduce the reader to the area of software development, Web-based systems, CAD/CAM and the sponsoring company. Problem statement and motivation for the research conclude this chapter. The structure of the thesis is explained at the end of Introduction.

1.1 Introduction to software development

The thesis discusses the development of Web-based CAD/CAM systems, that is software of a certain kind. Therefore understanding of software development basics is essential for this research.

"*Software* is a general term for the various kinds of programs used to operate computers and related devices" [1]. Software that is developed to help the user to perform specific tasks or to perform specific functions for another software is called *application software*, or just *application* [2]. Large and complex software are often approached with the application of systems theory in software engineering context, to allow focusing on the major components of software and their interactions [3]. In this regard, the term *software system* is often used as a synonym of software.

Software development refers to a "planned and structured process" of creating a software product [4], which does not necessary mean development of new programs, but also may incorporate re-engineering or making modifications to previously developed software, reuse of software components, software product maintenance, etc. [5, 6]. Software development is often used as generalisation of the notion of *software engineering*, but it does not necessarily include the engineering paradigm, i.e. "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software", as defined by the IEEE Computer Society [7].

The software development process may vary for different organisations and different kinds of software systems and depends on the applied software development methodology. *A methodology* establishes an approach to solving a problem, assuming a specific collection of principles and practices, with a defined set of rules, phases, tasks, methods, techniques and tools [8, 9]. Methodology defines the employed combination and timing of software development activities, feedback and control methods [6]. In turn, *software development method* refers to a specific technique for completing a particular software development task.

A description of a methodology could be divided in four parts: 1) identification of the problem area, that could be solved by applying the methodology; 2) providing guidance on actions and their timing for the successful application of the methodology; 3) advising of methods for ensuring the quality of deliverables; and 4) providing toolkit for the methodology [9].

A concept of software development methodology is concerned with notions of framework and model. A *framework* can be used by a methodology to "structure, plan, and control the process of developing a software system" [10] and provides the development team with a flexible guideline that could be adjusted for custom conditions [11]. A *model* establishes a design specification or a description for processes and methods, that can be implemented by a methodology or framework.

There are several basic approaches applied to software development [10]: *Waterfall*, *Incremental*, *Spiral*, *Rapid Application Development (RAD)*, *Prototyping* and *Extreme Programming* (a short comparison could be found in Table 2.1).

A wide variety of software development methodologies and frameworks have evolved over the years (see Section 2.4.1), continuously adopting to the trends in the software development industry and facilitating the development of various software projects in different circumstances. Every methodology or framework has its own recognised strengths and weaknesses and could be successfully applied to a limited scope of software projects with specific team size and expertise, organisational and technical characteristics [10].

1.2 Web-based applications

As the research focuses on Web-based CAD/CAM systems, which are a kind of Web-based applications, terms related to Web-based software are clarified.

The main feature of a Web-based CAD/CAM system that distinguishes it from the conventional desktop CAD/CAM software is the ability to work within the World Wide Web. This is enabled by utilising *Web technology* – a set of technical, communication and software methods for user collaboration using the Internet.

A *Web-based system* could be described as an application that is resident on a server and accessible over a network. The application could be partially of fully downloaded from the Web server to the client when executed. Two types of Web-based systems may be differed:

- *Browser-based* application refers to "a computer software application that is hosted in a browser-controlled environment (e.g. a Java applet) or coded in a browser-supported language and reliant on a common web browser to render the application executable" [12, 13].

- *Client-based* Web application employs "client/server" architecture and uses standard Web protocols for interactions between the server and the client. Client portion of the application could be stored on the client side or downloaded from the server each time the application is run, but it does not necessarily need a browser to operate [12].

The popularity of Web-based applications can be explained with the advantages of Web-based approach compared to the desktop software, which are reviewed in Chapter 2 (section 2.2.2). Web-applications bring different user experience to the software and require special approach in the development and maintenance.

1.3 CAD/CAM and CSE software

An introduction to the area of CAD/CAM and related terms are provided for the clear understanding about the research subject, because Web-based CAD/CAM systems is a particular case of CAD/CAM software. As for some CAD/CAM applications capturing knowledge for solving engineering problems might be challenging, an introduction is also provided to the field of Computational Science and Engineering (CSE).

Today most manufacturers employ computers in product design, analysis, and manufacture. "The application of digital computers in engineering design and production" is called *Computer-Aided Design and Manufacturing (CAD/CAM)* [14]. Computer-aided technologies allow to improve business flexibility, product quality, efficiency of personnel and utilisation of equipment, increase production volumes and profitability [14].

Computer-Aided Design (CAD) is the application of computer technology for the development of a product concept into a detailed engineering design, usually complemented with geometric models, which can be manipulated, analysed and refined [14]. Sometimes CAD is combined with *Computer-Aided Engineering (CAE)* capabilities and provides the ability to "estimate the performance and cost of design prototypes and calculate the optimal values for design parameters" [14].

Computer-Aided Manufacturing (CAM) is the application of computer technology for "converting engineering designs into finished products" [14] by providing assistance in process planning, supply of materials and production management, as well as control of various industrial equipment. Today, majority of new machine tools implement *Computer Numerical Control (CNC)* technologies, that enable "manufacturing complex shapes in an accurate and repeatable manner" [15].

Thus, summarising the described technologies, the term *CAD/CAM software system* in this thesis is viewed as a computer program that uses CAD tools create geometric models of the product that are then used by the CAM portion of the program to control the machining of the designed product.

Another term that should be viewed in conjunction with CAD/CAM technologies is *Computational Science and Engineering (CSE)*, which incorporates "the development of problem-solving methodologies and robust tools for the solution of scientific and engineering problems" [16]. "CSE is a broad multidisciplinary area that encompasses applications in science/engineering, applied mathematics, numerical analysis, and computer science" [16] (see Figure 1.1) and may require expertise in the application area, mathematical modelling, numerical analysis, algorithms, software development, analysis, validation and visualisation of results [16].

Similar to CSE software, CAD/CAM system development usually involves expertise in several areas, including mathematical modelling, algorithms, software implementation and design data visualisation.

CAD/CAM and CSE both have their own nuances, that could be best characterised with the following words of Frederick P. Brooks, the author of The Mythical Man-Month [18]: "A scientist builds in order to learn; an engineer learns in order to build". The

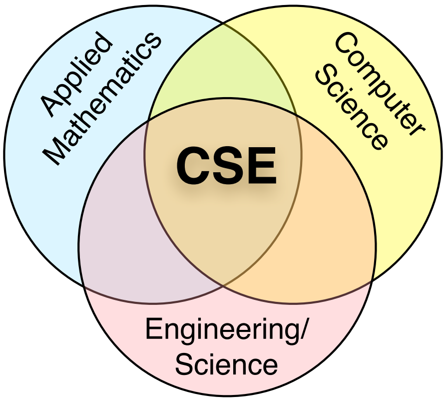


Figure 1.1: Multidisciplinary nature of CSE [17]

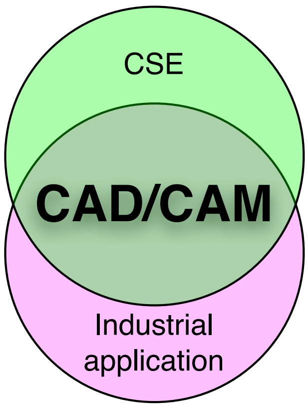


Figure 1.2: CAD/CAM incorporates industrial application and CSE features

purpose of CSE is mainly scientific discovery and extending knowledge in a particular field, whereas the purpose of CAD/CAM is application of the knowledge for solving practical problems. Although, capturing of this knowledge might be challenging for some CAD/CAM applications.

Another important aspect of CAD/CAM software is industrial application for producing real products, and consequently considerable attention to the robustness, usability and reliability of the results, produced by the system. The relationship between CSE and CAD/CAM could be presented as shown in Figure 1.2.

As noted, CAD/CAM systems serve to solve practical problems using accumulated knowledge. CAD/CAM could be beneficially applied to eliminate all sorts of production problems, but the successful application of the CAD/CAM technology would not be possible without applied knowledge and expertise in the specific domain.

1.4 Sponsoring company and KTP project

Dathan Tool & Gauge Co. Ltd. (Dathan) are specialists in the manufacture and supply of gear shaper cutters, rack type cutters, straight bevel generating tools, hobs and shaving tools for a variety of automotive, aeronautical and defence industries worldwide [19] (Figure 1.3). They are the only United Kingdom based manufacturer of gear shaper cutters and one of less than 10 reputable manufacturers in the world [20]. They export approximately 80% of all tools and supply most of the major aerospace automotive manufacturers. Their key strength are product quality, technical knowledge and support, and their ability to handle low volumes.

Figure 1.3: Gear cutting tools supplied by Dathan: a – gear shaper cutters, b – rack type cutters, c – straight bevel generating tools, d – hobs, e – shaving tools.

Gear cutting tool design and production involves extensive specialist knowledge that has been acquired and refined by Dathan over three generations [19]. The share of customised products produced by Dathan comes up to 40-50%, therefore flexibility of design and manufacturing is essential to ensure the capability to fulfil the requirements of any gear producer in the most efficient way [19]. Dathan uses CAD/CAM for rapid assessment of individual cutting requirements and optimisation of cutter characteristics to maximise its effectiveness, and applies computer controlled machinery for precise and consistent reproduction of designed cutting tools.

This research is a part of Knowledge Transfer Partnership (KTP) between Dathan and Cranfield University. KTP is a part government-funded programme helping British businesses to improve their competitiveness and productivity through the collaboration with universities and better use of knowledge, technology and skills [21]. This KTP project is in line with Dathan's recent strategic initiative to enhance its CAD/CAM capability by exploiting the latest advances in software and hardware technologies. The aim of the project is to develop software solutions, that will deliver the best possible support for Dathan to provide a flexible and rapid design and manufacture service required by its customers.

The reasons why Dathan is sponsoring this research include the following:

- The company is migrating from deprecated software program for gear shaper cutter design and looks for a future-proof software solution. The old software for gear shaper cutter design consists of separate design applications for each type of cutting tools and is running on Hewlett Packard workstations without support of modern data storage, retrieval and communication capabilities.

- Specialised production-ready CAD/CAM software for gear shaper cutters (GSCs) is not available in the market, and off-the-shelf general purpose CAD/CAM software packages have limited usefulness for gear shaper cutter design and manufacture [20].

- Implementation of new technologies promises to provide significant benefits for the industry in terms of world wide collaboration and improved software maintenance and manageability, utilising modern data storage and retrieval technologies.

1.5 Problem statement and motivation

Today, along with the manufacturing globalisation the need for collaboration between geographically distributed business branches and international customer involvement has increased. Large companies, that have distributed manufacturing and research and development sites with their customers scattered across the globe, are looking for tools to reduce operating costs and improve business processes while maintaining sustainability and expanding further in the global market. The Web is already used as a platform supporting many business areas and allowing to improve agility, responsiveness and overall competitiveness of companies all over the world.

This work is based on a hypothesis that Web-based approach provides significant advantages for CAD/CAM software development over similar desktop applications. Web-based applications have proven to be advantageous in many business areas (discussed in section 2.1.2) and could bring benefits (overviewed in section 2.2.2) to the area of CAD/CAM too.

The task of Web-based CAD/CAM software development has a number of associated challenges (listed in section 2.3.3), possibly able to eclipse the benefits of this approach. Therefore, the goal of this work is to find an efficient way of exploiting state-of-the-art Web technologies in the area of CAD/CAM.

1.6 Thesis content

This thesis is organised in nine chapters and has references listed at the end. The order of chapters is as follows: *Introduction*, *Literature Review*, *Research aim, objectives and methodology*, *Rationale and prerequisites for methodology*, *A methodology for creating Web-based CAD/CAM systems*, *Application of the methodology: case study 1*, *Application of the methodology: case study 2*, *Validation* and *Discussion and conclusions*.

The first chapter, *Introduction*, provides background to the key topics and motivation for the research. Section on *Thesis content* that describes the structure of the thesis is located at the end of the chapter *Introduction*.

The second chapter is dedicated to a review of literature, which includes discussion about the Web technology in industry, motivation for using Web-based approach in software development, review of Web-based approach application in product design and challenges associated with Web-based CAD/CAM system development, as well as review of software development methodologies and circumstances influencing the practical application of those methodologies. Section on identifying *Research gap* concludes the literature review.

Chapter 3 defines the aim of this research and sets objectives to guide the research process. The research methodology is described in this chapter to provide understanding of how the research aim and objectives are accomplished.

In Chapter 4 rationale and prerequisites for a dedicated methodology are discussed. The main concerns are associated with the development of science-intensive software to industry standards, user requirements elicitation, software design, project planning and change management.

A methodology for creating Web-based CAD/CAM systems is described in Chapter 5. The description of the methodology starts with defining the projects it can be applied to. After that the chapter lists key principles that the methodology is based on, followed by software development process model, planning approach and strategies for overcoming uncertainty during the project development. Discussion on Web-based CAD/CAM design concerns concludes the chapter.

The next two chapters are devoted to the application of the presented methodology in practice. The chapters describe two case studies. The first case study includes the development of a Web-based CAD/CAM system for involute spur gear shaper cutters. The second case study describes the development of a Web-based CNC code editor for online modification of a profile for producing gear shaper cutters. Challenges faced during the development of these case studies and implemented solutions are discussed at the end of Chapter 6 and Chapter 7.

Chapter 8 provides validation of the developed software by means of running real design examples and use cases, summarising industrial feedback, as well as performing analysis of the Web-based CAD/CAM software advantages and limitations.

Finally, the results of the conducted research are summarised in the last chapter, which presents conclusions and key findings, discusses limitations of the present work and establishes future research directions.

1.7 Summary

The Chapter provides understanding of relevant terms and principles in the areas of knowledge underlying CAD/CAM software development. The subsequent sections in this chapter introduce the reader to the area of software development, Web-based systems, CAD/CAM and the sponsoring company. Problem statement and motivation for the research conclude this chapter. The structure of the thesis is explained at the end of the Chapter.

As the thesis discusses the development of Web-based CAD/CAM systems, that is software of a certain kind. Therefore understanding of software development basics is essential for this research. Introduction to software development provides understanding for such terms, as software, application, software system, software development, software products, software engineering, methodology and other.

In the focus of this research are Web-based CAD/CAM systems, which are a kind of Web-based applications. Therefore terms related to Web-based software are clarified in the introduction to Web-based applications. Browser-based systems that run within the users Web browser and client-based systems that resemble local applications are explained.

An introduction to the area of CAD/CAM and related terms are provided for the clear understanding about the research subject, because Web-based CAD/CAM systems is a particular case of CAD/CAM software. As for some CAD/CAM applications capturing knowledge for solving engineering problems might be challenging, an introduction is also provided to the field of CSE.

The Chapter also introduces the company sponsoring this research and KTP between the sponsor company and Cranfield University, as the research is a part of this project. The KTP project is in line with sponsor companys strategic initiative to enhance its CAD/CAM capability by exploiting the latest advances in software and hardware technologies. The aim of the project is to develop software solutions, that will deliver the best possible support for the sponsor company to provide a flexible and rapid design and manufacture service required by its customers. The reasons, why the company is sponsoring this research are listed in Section 1.4.

Problem statement and motivation for the research are provided in this Chapter. This research is based on a hypothesis that Web-based approach provides significant advantages for CAD/CAM software development over similar desktop applications. Web-based applications have proven to be advantageous in many business areas (discussed in section 2.1.2) and could bring benefits (overviewed in section 2.2.2) to the area of CAD/CAM too. The task of Web-based CAD/CAM software development has a number of associated challenges (listed in section 2.3.3), possibly able to eclipse the benefits of this approach. Therefore, the goal of this work is to find an efficient way of exploiting state-of-the-art Web technologies in the area of CAD/CAM.

Chapter 2

Literature Review

To solve the problem, stated in Section 1.5, first of all it is required to confirm the ability of World Wide Web (WWW) to serve as base technology for CAD/CAM software and to be advantageous over similar desktop applications. For this purpose an insight into the role of Web technologies in industry is provided to understand the level of Web technology maturity and confirm the ability of Web to handle CAD/CAM applications. Common trends in software development and reasons for increasing popularity of Web applications are investigated to determine if the development of Web-based CAD/CAM software is inline with current software developments. Therefore, Web-based CAD/CAM software advantages are listed and possible benefits for the industry are outlined.

The experience of applying Web technology for product design is reviewed and implementation technologies are analysed to find any common approaches that could be used to develop a systematic approach for creating Web-based CAD/CAM software. The challenges associated with the development of this kind of software are also discussed.

To find an efficient way of applying Web technologies in the area of CAD/CAM, existing software development methodologies and methodologies relevant to the research are reviewed. Application of software development methodologies investigates the circumstances influencing their practical application.

Finally, based on the study of literature, this Chapter identifies the research gap, which will guide this research.

2.1 Web technology in industry

Web technologies rapidly developed as the Internet become more popular and now it is hard to imagine an area, that remains untouched by WWW. According to the Netcraft

Web Server Survey in April 2012 there were nearly 677 million distinct websites on the Internet, and about 190 millions of them being active Web servers [22]. Such involvement of human society into the Web also influences the development of industrial applications. Advantages offered by Web technologies make them attractive for different business areas, and make software developers look for solutions that allow to utilise these advantages.

This section provides an overview of Web technology and its application areas with the aim to create understanding about the level of readiness of the WWW to serve as base technology for the development of complex industrial software systems such as CAD/CAM applications.

2.1.1 Web technology overview

Originally designed with an idea of accessing interlinked documents located on different computers across the Internet [23], WWW has evolved from static through dynamic to interactive environment, which is able to handle large applications with complex business logic [24]. Figure 2.1 presents a timeline of some basic World Wide Web Consortium (W3C) activities created from 1994 up to 2011 (based on [25, 26]), which reflects general trends in Web technologies development.

A noticeable characteristic of the evolution of the Web is that innovation often results in technology fragmentation, e.i. changing in conflicting directions [24]. In this context in order to enable maximum flexibility and reach Web application developers should be aware both of new technological developments and well supported standards.

Research of trends in Web application development reflects that Web applications evolution started with thin-client architecture and found its continuation in transferring more and more processing tasks from the server to the client [27]. This process was accompanied by improving interactivity and client-side processing capabilities of Web applications by introduction of JavaScript, caching, Asynchronous JavScript and XML (AJAX). Later, many browsers were equipped with graphics sub-system, which enhanced Web application capabilities in creating and processing content that previously was generated on the server [28].

As Web application functionality covers more everyday tasks, the requirements for their safety, performance and reliability increase. To address the associated challenges Web engineering offers approaches applicable for the design and development of Web-based software, such as service-oriented systems, model-centric architecture, service-based

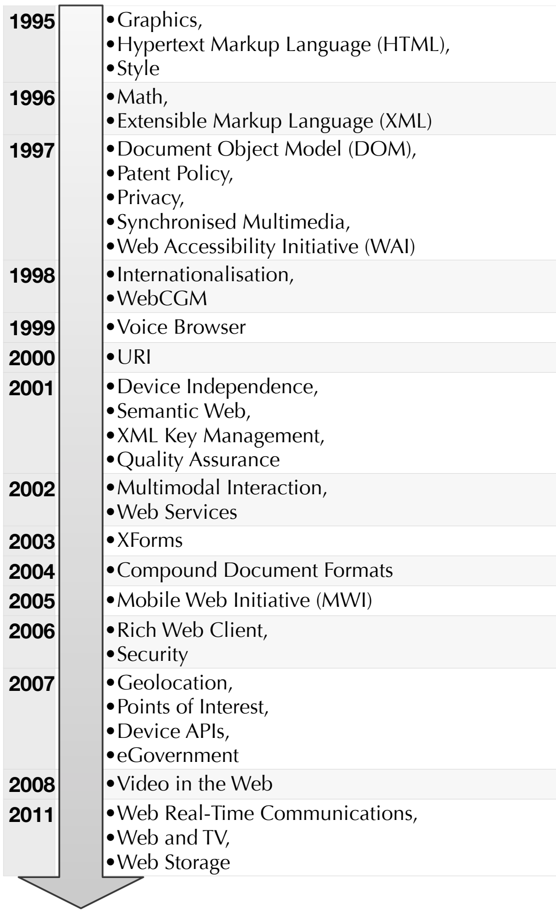


Figure 2.1: W3C activity creation timeline from 1994 up to 2011 (based on [25, 26])

mediation, patterns, component-based deployment, agile methods, prototyping, etc. [24, 20].

2.1.2 Application areas

Researchers' interest about Web-based systems has grown with time. Figure 2.2 shows increasing number of publications about Web-based systems from 1997 to 2008.

Today, competitiveness of manufacturing companies rely on their ability to respond to customer needs and bring products to market faster, reduce inventory and production costs and avoid production problems related to inventory and materials. Achieving these goals is product development task, involving speed, agility, responsiveness and knowledge [29].

Product Data Management (PDM) systems play an important role in agile product development. Web-based approach to the development of PDM systems ensures better

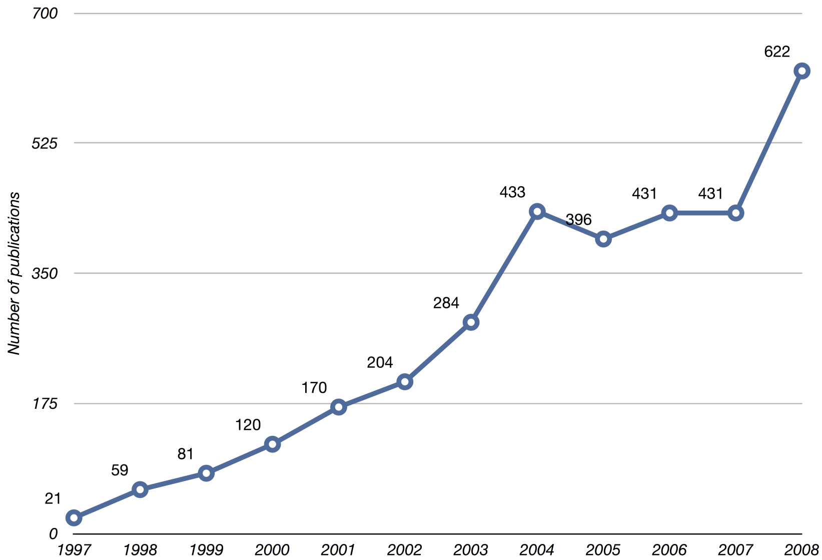


Figure 2.2: Number of publications about Web-based systems

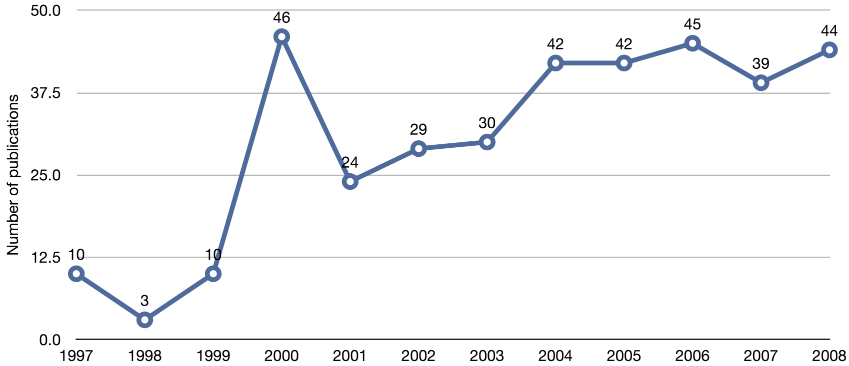


Figure 2.3: Number of publications about web-based PDM

adaptability and flexibility than conventional approach. Applying Web technology to PDM infrastructure provides such benefits, as better user-friendliness, better accessibility and applicability, improved linking to the supply chain and coordination between geographically distributed organisations, etc [30]. Figure 2.3 indicates increased number of publications about Web-based PDM after a splash of interest to the research in this field in 2000.

Along with the progress in Web technologies and design methods of Web-based applications, research in the field of Web-based Decision Support Systems (DSSs) has grown significantly over the recent years. Figure 2.4 reflects the number of publications about Web-based DSSs from 1997 to 2008.

It is interesting to remark, that Web-based DSS research covers many different subject areas and apart from computer science there are many publications (22%) in engineering.

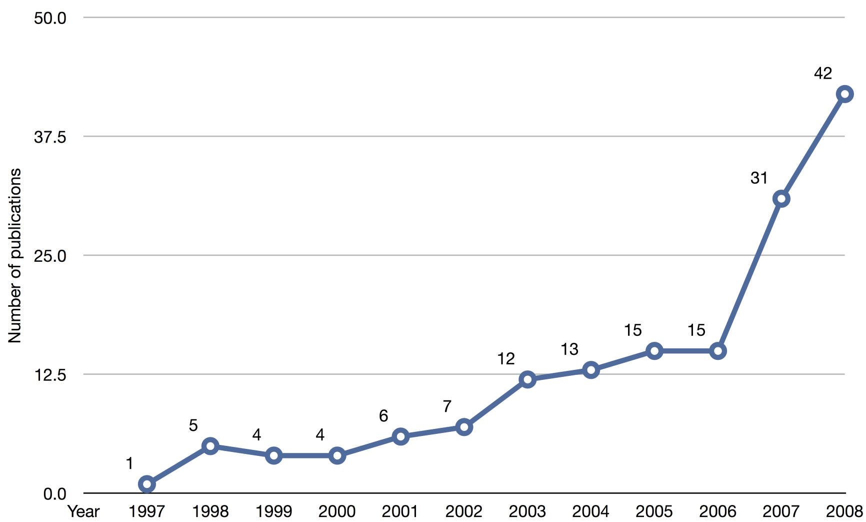


Figure 2.4: Number of publications about Web-based decision support systems

Figure 2.5 shows breakdown of all publications about Web-based DSS from 1997 to 2008 by subject areas.

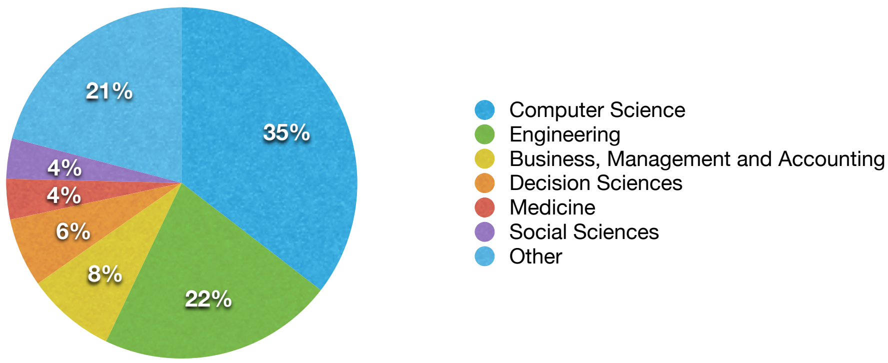


Figure 2.5: Subject areas of Web-based DSS research

Significant part of Web-based DSS research devoted also to such areas, as business, management and accounting, decision science and medicine. Advantages from the use of Web technologies in this areas encourage researchers and software developers to find solutions for applied Web-based systems [20].

2.2 Motivation for Web

To understand how creating of Web-based CAD/CAM applications is aligned with common trends in software development and estimate possible benefits for the industry from implementation of this kind of systems, this section provides an insight into past and present software development practices and summarises arguments in favour of Web-based software development.

2.2.1 Software development trends

Historically, programs have been developed as stand-alone applications. Connecting separate computers in networks resulted in development of applications that communicate with each other and share data over the network. As the Internet has grown, communications over the global network increased.

Moreover, the Internet is becoming the computer software platform, that eliminates application software dependance on the type of operating system or computer manufacturer. As an indicator here appears invention of Application Service Provider model, where instead of owning the software, the user accesses the application hosted on the developer's server over the Internet as required and pays for the service [31].Web is often used for delivery of software to the end user by means of integrating the existing software package with the Web environment or extending it with a module enabling to use this software over the Web. The vast majority of modern enterprise applications have built-in Web-publishing option. And already proved efficiency of Web mashup model for the creation of composite enterprise applications [32] indicates broad opportunities for integration, better information utilisation, business interoperation and collaboration.

As significant amount of functionality has been created and shifted to the Web; Web browsers are quickly becoming a dominant client application platform. Recent research undertaken by computer scientists from Microsoft Research intended to address a "paradigm shift" in how personal computers are used. Scientists present a vision of new computing paradigm, which enables an application model that synthesises the best elements from both desktop and web applications [33].

2.2.2 Advantages of Web-based approach

Distributed implementation of client-server architecture characteristic to Web-based systems enables centralised configuration, maintenance and control, this way making Web-based systems more manageable and reliable than stand-alone software [20]. Application virtualisation and session virtualisation technologies, like Citrix XenApp, Microsoft App-V, VMware ThinApp and similar, witness to an existing demand for server-based computing software and associated advantages, that could be applied in many areas, including product design and management [34].

Web-based systems win over desktop applications in scalability, portability and stability, which is capacitated by abstracting from client platform and thus reducing hardware and software conflicts and making the Web-based system less prone to crashing and creating technical problems [20, 35]. "In contrast with desktop programs Web-applications require less Random Access Memory (RAM) on the client side, they could ensure better data safety and availability from any place with the Internet connection" [20].

Taking into account, that direct costs and total cost of ownership of server-based computing usually turns out cheaper compared to desktop deployments [36], preference of Web-based systems over stand-alone applications is financially motivated.

Web-applications could be attractive for developers with background in product design and manufacture and little expertise in software development, because even without deep knowledge of different operating systems and software packages it is possible to ensure cross-platform compatibility and performance of Web applications [20]

Experimental Web-based software for design and manufacture demonstrated better accessibility, user-friendliness and processing time in a usability research [37], and collaborative design advisory systems based on Web technologies are reported to increase the design efficiency [38].

2.3 Web-based approach in product design

Web technologies could be used to improve many aspects of product design and development, including product data management, design collaboration and design data communications, customer engagement and business communications. Web-based product design frameworks attempt to address collaboration challenges in distributed and heterogeneous environments [39], as well as solve data inconsistency problems by avoiding repetitive data input due to integration of CAE, CAD, PDM and Enterprise Resource Planning (ERP) systems [40]. Many commercial solutions trying to address problems in traditional production, associated with integration in product cycle, database and commercial package levels [41].

Web-based product data management systems facilitating product development and related communications are offered by majority of CAD companies - including Autodesk, Karat Software, Lectra [20]. Implementation of Web technology enhances collaboration and distribution capabilities of Product Lifecycle Management (PLM) [42]. Solutions, like ThomasNet's 2D/3D WebCAD Publishing Technology, enable utilising WWW infrastructure advantages by providing the ability to publish product prototypes on company's website, where they are available for input into one of the major CAD system [43].

Today the Internet could be even used as a platform for delivery of CAD services, like in the case with Microwave CAD introduced by P. Sypek et al. [44] And with the advances in Web technologies CAD/CAM finds its own new niches on the Internet [20].

The way engineers work today is significantly impacted by social-networking technologies, such as wiki, blogs, and RSS feeds. Tying together design discussions about products, that are usually distributed over emails, blogs and instant messages would help to improve collaboration and allow to keep the underlying engineering intent better. And some of CAD developers are already working in this direction and starting to build social-networking capabilities into their software [45]. Many of major CAD developers are experimenting with 3D virtual worlds in order to find out how it might be used for creating next-generation CAD software.

In this section various Web-based CAD systems are overviewed and classified in order to focus further attention on entirely Web-based CAD systems. Web-based CAD/CAM software implementation methods and technologies are researched to understand the level of maturity of Web technologies and confirm the ability of WWW to handle this kind of applications. It would be hard to overestimate the value of the previous experience in Web-based CAD/CAM software development, that's why the research is based on real solutions described in the literature.

2.3.1 Classification of Web-based CAD/CAM systems

Distributed CAD/CAM systems could be classified depending on the implementation of Web technologies as follows:

- *Built with Web* – these are Web-based CAD/CAM systems with client-server architecture integrated with proprietary CAD software:
  - *Using off-the-shelf CAD/CAM software on the client side*, like SmartFab that uses SolidWorks at front end as design User Interface (UI) [37] or like Web-based collaborative visualisation system with CATIA software in the core of its Design Tool module [46].
  - *Using off-the-shelf CAD/CAM software on the server side.* Examples from the literature include E. Ivask's et al. Web-based framework for distributed remote laboratory integrated with digital circuits CAD [47]; intelligent design

system for injection moulds integrated with SolidWorks [48]; or Seo's et. al. Web-based CAM system that uses PosCAM for CNC code interpretation [49].

- *Built for Web* – entirely Web-based CAD/CAM systems:
  - *Web-based CAD/CAM systems* that use special environments or viewers for design visualisation; these systems are implemented using either Java technologies (such as Java Run-time Environment (JRE)), Flash, X3D or Virtual Reality Modelling Language (VRML). "This approach has known limitations, when client is bound to use certain software to enable full functionality".[20] P. Sypek's et al. microwave CAD [44], Autodesk Homestyler (homestyler.com) and P. Guan's et al. Web-based virtual grinding machine tool system were developed using these technologies.
  - *Browser-based CAD/CAM systems* do not need any additional tools on client side, but a Web-browser. These applications are highly compatible. There are known few examples implemented using SVG or Web-browser canvas for design data visualisation: parametric Web-CAD for box culvert design [50], Web-CAD (google AutoCAD) [51], Planner 5D (planner5d.com) and others.

Creating entirely Web-based CAD systems forms the main interest of this research, as this kind of CAD applications allows for better possibilities for utilising Web technology advantages.

2.3.2 Web-based CAD/CAM implementation methods and technologies

Research of Web-based CAD/CAM systems implementation methods and technologies is based on the analysis of a number of developed Web-based CAD/CAM solutions (Planner 5D, Sweet Home 3D, IKEA planning tools, Mydeco 3D Planner, 3DTin, Autodesk Homestyler, Aftercad online) and Web-based CAD/CAM applications described in literature ([52, 53, 54, 55, 56, 57, 58, 41, 59, 48, 46, 60, 61, 62, 63, 64, 44, 65, 50, 66, 67, 68, 69, 70, 47, 71, 72, 73, 74, 37, 75, 76, 49, 49, 51, 77, 78]). The Web-based CAD/CAM systems under review were developed starting with 2000.

Considering classification given in section 2.3.1 a big part of surveyed CAD/CAM software is integrated with proprietary CAD/CAM package being developed with the aim to enable some of advantages of Web technologies for the existing CAD/CAM software.

Figure 2.6 shows Web-based CAD/CAM software breakdown into separate and integrated Web-based applications, where 28% from all software are solutions using offtheshelf CAD/CAM software either on the server, or on the client side.

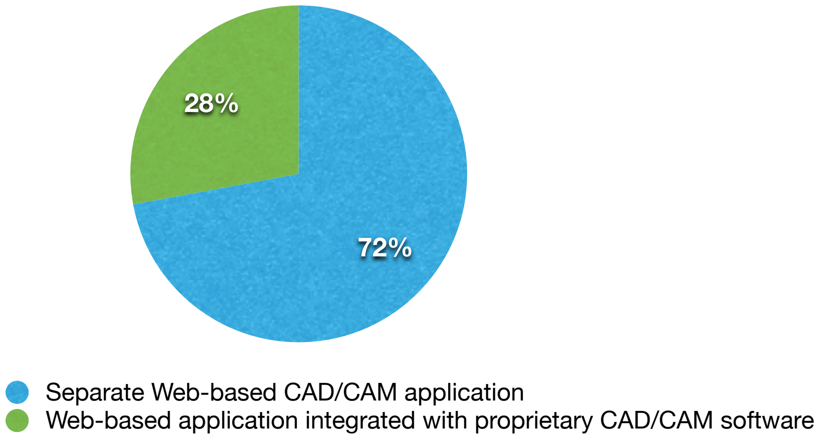


Figure 2.6: Percentage of separate Web-based CAD/CAM solutions versus integrated with proprietary CAD/CAM software

As follows from a diagram presented in Figure 2.7, 49% of developed Web-based CAD/CAM applications are not production systems and therefore could not be used within industrial environment. This finding proves claims of other researches [79], that most of Web systems in product design and manufacture are developed as proof-of-the-concept prototype Web applications.

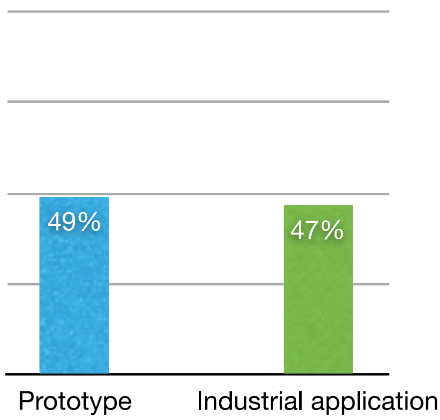


Figure 2.7: Prototype software versus industrial Web-based CAD/CAM applications

Looking at Figure 2.8 one could make a conclusion that the vast majority of Web-based applications in product design and manufacture utilise Java technology. Indeed 63% of all surveyed Web-based CAD/CAM applications use Java at greater or lesser extent and there is a reason for that. Java and Java3D Application Programming Interface (API) used in 3D applications provide broad capabilities for visualisation, dynamic interactions and collaborative access management [61, 70].

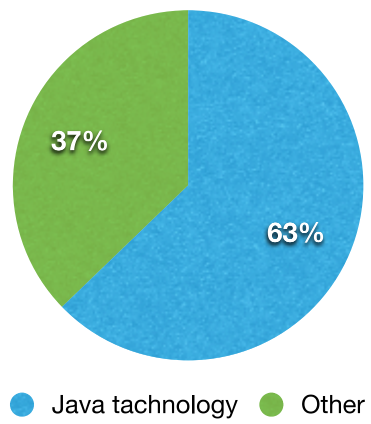


Figure 2.8: Percentage of Web-based CAD systems implementing Java technology

Figure 2.9 presents a breakdown of visualisation technologies, implemented in Web-based applications in product design and manufacture. The two leading technologies are VRML, which is used in 32% of all applications, and Java3D, utilised by 27% of Web-based CAD/CAM systems. The fifth revision of the Hyper Text Markup Language standard (HTML5) technology has been utilised in a number of Web-based CAD/CAM projects developed over the recent years, which makes 14% of all surveyed applications.

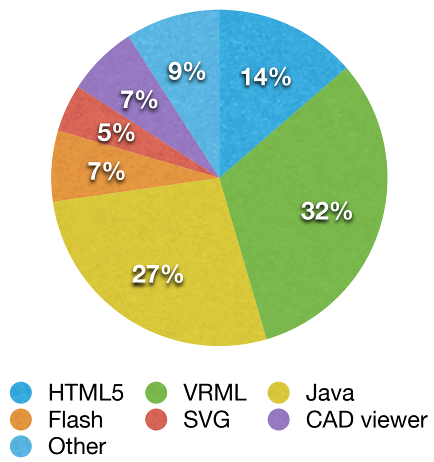


Figure 2.9: Visualisation technologies used in Web-based CAD systems

Such a big part of integrated solutions and Java-based implementations amongst Web-based applications in product design and manufacturing could be explained though risks, which arise in projects, characterised with high level of novelty and extended time frame. Rapid development and relative instability of technologies makes the task of Web-based CAD/CAM application more challenging [80].

The following conclusions could be made while summarising the reviewed Web-based CAD/CAM methods and technologies:

- Using integrated solutions and well known programming technologies allows minimising risks associated with novelty and complexity of the project;
- Integration with proprietary CAD components limits the extent of utilising Web-based technology advantages in terms of flexibility, compatibility, openness and ease of maintenance;
- Despite the plethora of Web-based technologies there are no defined generic approach for the development of Web-based CAD/CAM systems;
- There are no defined system architecture and development methodology for complex Web-based CAD/CAM applications purposed on using in production.

2.3.3 Challenges of Web-based CAD/CAM development

Running software over the Web is associated with a number of challenges, such as internet connectivity and security issues, interactivity requirements and graphics visualisation, user collaboration, and of course application architectural decisions and performance optimisation.

2.3.3.1 Internet connectivity and security

Web-based CAD/CAM systems have the same security risks, that are common to other Web applications [81], and the measures for elimination or mitigation of those risks are similar [82]. As the data, that Web-based CAD/CAM applications transmit over the Internet as plain text might be business sensitive, it is important to protect these data from unauthorised access. While this particular issue could be resolved by accessing the application using protected corporate network, Virtual Private Network (VPN) or by using Secure Sockets Layer (SSL), the design of Web-based CAD/CAM application should take into account both technical solutions for elimination of potential security threats, and possible processual flaws [20].

Web-based applications rely on data transfer between the server and the client and their performance depends on the network bandwidth and intensity of traffic, which in turn depend on the number of users working with the Web-based application simultaneously. Minimising the amount of data transferred over the network is an important concern for highly distributed Web-based CAD/CAM systems.

"Persistent Internet connectivity is essential for Web-based applications in general and critical for such highly interactive applications as CAD software in particular" [20]. It is possible to reduce the risk of data loss due to connectivity interruptions by implementing capabilities for saving an opened design data automatically or support of working offline and synchronising the data with the server upon getting back online. Companies, that use Web-based CAD/CAM systems in their work should insure network infrastructure capable to handle operation of the implemented system.

2.3.3.2 Interactivity and graphics visualisation

Despite the importance and challenge of design data visualisation in Web-based CAD/CAM systems, the literature describes rather uniform solutions, which rely on add-on components, like VRML and Java, for the implementation of the majority of industrial Web-based CAD/CAM applications. Visualisation of vector graphics in Web-browsers is lacking a common approach: "Scalable Vector Graphics (SVG) was approved by W3C as a standard for vector graphics on the Internet", nevertheless it "does not appear to be used widely for technical drawing visualisation" [20]. The most recent Web applications developed to solve CAD tasks tend to employ HTML5 technology and Web Graphics Library (WebGL).

Geometry modelling faces challenges when dealing with heterogenous systems and three-dimensional data exchange, "especially under the narrow band network condition" [83].

Analysis of Web-based solutions for sharing product and manufacturing data explored the profusion of proprietary and open source standards developed in the absence of a unified 3D Web standard, which causes compatibility problems when exchanging information over the Web [84].

2.3.3.3 Sharing tasks between the server and the client

Performance of sophisticated Web applications heavily depends on the way how tasks are distributed between the server and the client. Determining what portion of the software would be appropriate for accomplishing certain tasks should be based on the nature of the application problems [79]. Also when dealing with large scale Web-based applications, where users are distributed over the world, performance improvements could be achieved with job sharing among multiple geographically distributed servers. While designing Web-based CAD/CAM software systems it is essential to consider the level of required interactivity, overall system performance and number of simultaneous uses of the same application server.

2.3.3.4 User collaboration

Collaborative product development could be described as a recursive process where two or more people, whether it would be designers or customers, are working together on product development. Collaboration could include three aspects: the geographical, disciplinary and time aspect of collaboration.

To provide capability of collaborative work of multiple users Web-based applications have to be designed and developed accordingly. "Integrity and consistency must be maintained between participants as the collaborative design progresses although local and temporary inconsistency is both allowable and necessary to enable overall optimisation" [85].

Workflow management is another challenge in Web-based collaborative product development due to simultaneous access of multiple users to Web pages containing hyperlinks and therefore allowing users browse freely to any location. "Accessing information on the Web is unidirectional, asynchronous and limited by simple client-server model in which only predefined data are provided" [85]. In contrast, handling geographically distributed collaborative product design by the WWW requires "a bidirectional, synchronous Web-based design tool" [85]. For the implementation of collaboration and interactivity features of Web-based CAD/CAM systems synchronous and asynchronous communication methods are available and each should be used when appropriate [86].

2.3.3.5 Design and development

Difficulties in the design, development and implementation of Web applications in product design and manufacture have been discussed in the literature [79]. Considering that most of those applications are developed as proof-of-concept prototype Web applications, main issues associated with Web-based CAD/CAM software design and development include (based on [79]):

- Selecting an appropriate approach for Web-based application development among the plethora of software development practices;

- Employing an information medium intended for sharing electronic documents for handling interactive and complex CAD applications is challenging;

- Rapidly changing Web technology, which makes it difficult to follow especially for people with background in product design and manufacture, who's expertise in software development is limited.

Many prototype Web applications were developed "for studying the feasibility and potential of Web technologies and advanced manufacturing" [86], yet many Web-based CAD/CAM software development challenges have no firm solutions and "developers have to rely on their experiences in designing, implementing and deploying web applications" [20, 87].

Advancements and spreading of Web technologies, despite growing software complexity, led to introduction of development approaches with reduced time to market. "The increased focus of developmental efforts on tools seems to have led to some reduction of emphasis on serious analysis, modelling, design, and process adoption. This in turn can adversely affect the robustness, effectiveness, change management and predictability of the applications so developed" [80].

2.4 Software development methodologies

As it follows from section 2.2 development of Web-based CAD/CAM systems could be very promising solution for many business areas. Due to the challenges, associated with development and running this kind of software over the Web, providing an effective web-based CAD/CAM software development methodology is essential for enabling Web-based approach benefits in industry.

This section provides overview of evolution of software development methodologies, reviews methodologies relevant to the research, investigates circumstances influencing practical application of software development methodologies, compares the most common software development models and summarises positive and negative software engineering experiences.

2.4.1 Evolution of software development methodologies

Following the trends in software engineering from the 1950s to the present, the approach to software development experienced several transformations. A study of the 20th and 21st century software engineering [88] tried to apply Hegel's hypothesis to explain the evolution of software engineering practices as a path of thesis, antithesis and synthesis, that humans undertake in the process of increasing understanding about a subject. It started with the adopting of hardware engineering practices, but as software is much easier to modify than hardware, the approach changed more to software crafting in 1960's, counterpoising a "code and fix" practice to a rigorous sequential process of 1950's. Increasing scale of software projects and the demand for more disciplined practices opposed the intuitive development by returning to sequential approach and adoption of waterfall process, as well as methods structuring and formalisation. The next decade proposed a synthesis of previous experiences for better productivity and scalability, developing Object-Oriented Programming (OOP) techniques, software factories, standards and maturity models in order to improve software evolvability and scalability, productivity and process compliance. 1990's presented concurrent development process as an antithesis to previously implemented sequential methodologies, emphasising software time-to-market and establishing development methodologies supporting concurrency control. In 2000's, further increase in pace of change of information technologies led to emergence of value-based software engineering and agile methodologies, prioritising usability and total ownership cost-benefits [88].

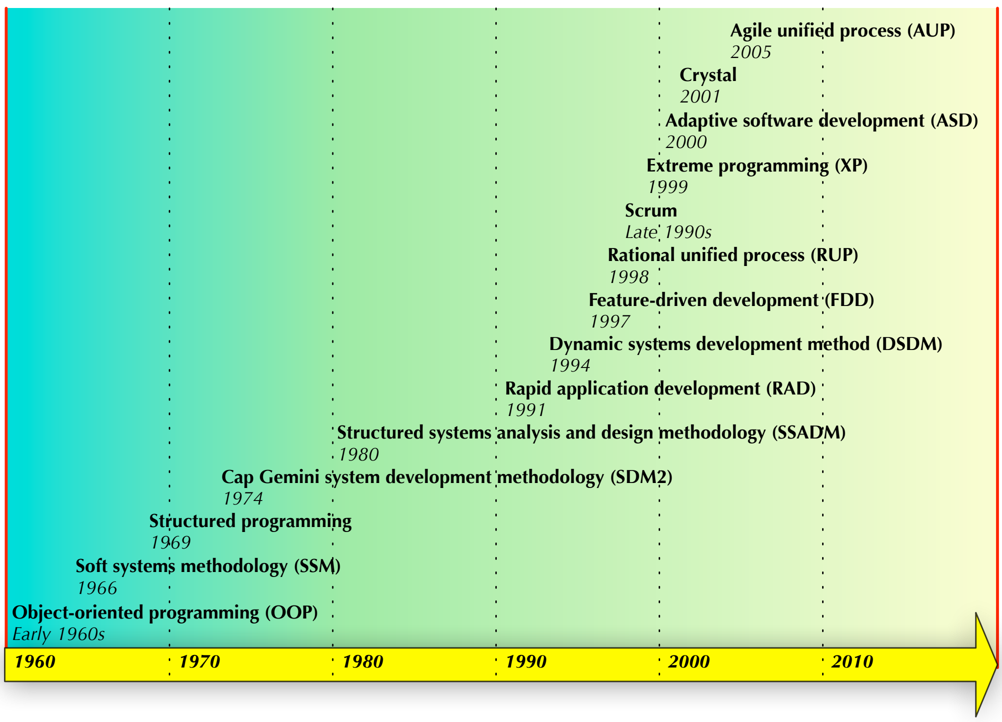


Figure 2.10: Evolution of software development methodologies (based on data from literature [89, 90, 91, 92, 88, 93, 94, 95])

Figure 2.10 shows the emerging of specific software development methodologies in a historical timeline (based on data from literature [89, 90, 91, 92, 88, 93, 94, 95]). Along with the increasing complexity of tasks, that were solved using computers, and therefore increasing complexity of the software, programming practices and software development methodologies were evolving. In 1970s software application size and complexity reached the level, when earlier common intuitive development could not meet practical requirements anymore. Applications became too large and complex to be normally maintained. Among the first measures addressing the problem of systematising software development process and application structure were OOP and structured programming. Introduction of Soft Systems Methodology (SSM) provided an approach for managing disordered real-world problem situations lacking formal problem definition [89, 90].

Many of the earlier popular software methodologies utilised a waterfall model, like System Development Methodology (SDM) or Structured Systems Analysis and Design Methodology (SSADM). The latter represents a good example of the rigorous document-led approach to system design. Cap Gemini System Development Methodology (SDM2), a revised version of SDM focused on solving the problem of meeting customer requirements, could be viewed as one of the early forerunners of the rapid application development methodology, highlighting the importance of cyclic design and implementation phases.

Lightweight software development methodologies appeared in the mid-1990s with the intention to eliminate drawbacks of heavyweight methodologies, criticised for being heavily regulated, regimented and micromanaged. Early lightweight methodologies include Dynamic Systems Development Method (DSDM), Feature Driven Development (FDD), Scrum and Extreme Programming (XP) [91, 92].

In 2010's the increasing role of the Software-Intensive Systems of Systems (SISOS) as a result of global connectivity raise new challenges for systems and software project processes reconsideration within the enterprise environment [88]. SISOS engineering and management principles emerged in 1990's and early 2010 support organisations by providing enterprise architectures, such as International Business Machines Zachman Framework, Reference Model for Open Distributed Processing (RM-ODP), Federal Enterprise Architecture Framework and commercial ERP packages [93, 94].

Contemporary software development methodologies usually are based on a concept of agile software development that assumes "iterative and incremental development, where requirements and solutions evolve through collaboration between self-organising, cross-functional teams. Agile methodologies promote adaptive planning, evolutionary development and delivery, time boxed iterative approach and encourage rapid and flexible response to change" [96].

2.4.2 Methodologies relevant to the research

In the absence of a dedicated methodology for the development of Web-based CAD/CAM software, it might be useful to review the experience in developing applications with some of the features, characteristic to this kind of software, in particular, methodologies used for the development of Web-based applications, CAD/CAM systems and scientific CAD/CAE and complex CSE software.

Reviewing software methodologies relevant to the research will help to find out methods that are effective for overcoming certain challenges of Web-based applications, CAD/CAM and CSE software, that could be combined or adapted to approach the development of Web-based CAD/CAM systems in a systematic and disciplined manner.

2.4.2.1 Web application development

Practitioners tend to remark a few limitations that are more explicit in Web application development compared to traditional software development, such as shorter time frames, variations in employees experience levels, lower level of user understanding, faster pace of technology change and the implementation narrowed to a single user interface (browser) [97]. Complemented with continuous application evolution, stringent security and privacy requirements, content-driven implementation with emphasis on visual design and presentation, integration of content and procedural processing, and performance and responsiveness influenced by the organisation and presentation of the content, all these characteristics of Web applications make their development different from other software. Thus a background is created for adopting, adapting and building dedicated Web application development methodologies, addressing the associated challenges [24].

Agile software development methodologies, such as XP, Scrum, timebox development, FDD, are addressing many of the challenges in Web development, but unfortunately there is not a complete and effective strategy for developing Web-based software. Many methodologies are lacking an effective approach to project effort planning, requirements gathering, visual design, testing and deployment, making these tasks even more challenging for large scale sophisticated Web-based software development projects. Nevertheless, it is strongly advised to use a systematic and documented approach to enable better control over the development process of large, high-performance and mission-critical Web applications [24]. It is also very important to make clear understanding of Web application usage context, to create such an architecture, that will be able to support the whole life cycle of the application, its evolution, maintainability and growth.

2.4.2.2 CAD/CAM software development

In accordance with existing practices [98] the development process of CAD/CAM software emphasises close customer partnerships and is normally split in several consecutive phases, each pursuing its own objectives. The development process of industrial CAD/CAM systems has some unique features, in particular it includes elaboration phase following after the inception phase defining project requirements, initial concepts and scope.

The primary objective of the elaboration phase is investigating of the best way how the project requirements could be met. In this phase any available software components that could be reused are identified, designs of the software architecture and user interface are created and proof-of-concept prototypes developed to assess the design and selected off-the-shelf components and develop strategies for mitigating major risks [98].

The development focuses on the correct software architecture on purpose, as the architecture has great impact on overall scalability, performance and maintainability of the CAD/CAM system. It is essential to deal with the biggest risks first and any unknown technologies involved in the project have to be investigated early in the project, as well as key scenarios have to be specified and implemented first. The architecture of the CAD/CAM system is viewed as a foundation, on which the rest of the application can be built further [98].

2.4.2.3 Scientific CAD/CAE and complex CSE software development

Development process of scientific CAD/CAE and complex CSE software has been previously discussed in the literature [99], and many aspects of agile software development practices discovered to be highly relevant for explorative science-intensive engineering software projects [100].

The development of scientific CAD/CAE and complex CSE software is associated with a number of challenges governed by the application complexity and science-intensive nature of these projects. The development process may be significantly stretched over time, that complicates introducing of new features and technologies and may require significant architectural changes. Thus, as in the case with industrial CAD/CAM systems, task allocation during the development of scientific CAD/CAE applications is based on task priority and complexity. For the overall quality of the product the number of complex and time consuming tasks scheduled for the development should descent to the end of the project.

As it was mentioned, the development of the science-intensive software is complicated due to requirements containing specific scientific information. To address this, it is advised to allow for initiation of theoretical researches and in depth problem investigation by field experts [99]. Reviewing the research progress and communicating its results to the involved parties. Adequate and up to date project documentation is vital for eliminating any misunderstandings should they arise among the developers throughout the project. One of the best ways to achieve this is to make every project team member responsible for maintaining documentation relevant to this part of the process, while the project manager accumulates and communicates all generated project documents.

Another important aspect of scientific CAD/CAE and complex CSE software development is that in contrast to the practices, common in software development industry, the working team in science-intensive projects usually is not a group of experienced developers, but rather is created specifically for the development of a particular software.

2.4.3 Application of software development methodologies

Different software development projects have different circumstances, thus the development itself should be approached accordingly, as what is good for one kind of projects is not necessary as good for another kind of software projects. Table 2.1 provides a short comparison of strengths and weaknesses of the most common software development models, as well as description of situations, where these models are most and least appropriate. The comparison is done based on the analysis of software development methodologies available in the literature [10, 6].

The comparison shows, that every software development model can be successfully applied to a limited scope of projects, obeying certain conditions. The choice of software development methodology hinges primarily on two factors: organisational environment and the nature of the application [6]. The relationship between these two factors defines the appropriate approaches to system analysis, design, development and implementation, putting different emphasis in these processes depending on the stability, knowability, and predictability of systems [101]. According to the proposed categorisation [101, 102], the relationships between a system and its organisational environment are identified as follows:

- *The Unchanging Environment*, which presumes unambiguous, complete and accurate definition of requirements and could benefit from the application of formal approaches, such as the Waterfall or Spiral Models.
- *The Turbulent Environment* with continuously changing system requirements, where the development could succeed from the use of methodologies, incorporating rapid development, throwaway prototyping, creating a highly modular design and reusable code.
- *The Uncertain Environment* characteristic to the development of highly innovative and pioneering systems, where it is not possible to accurately define requirements of the system ahead of time. The appropriate development methodologies emphasise learning, experimental process models, prototyping and rapid development.
- *The Adaptive Environment* characteristic to expert and teaching systems, incorporating possibility of requirements change initiated by environmental change in reaction to the system being developed. The development methodology of this kind of systems should support adaptation and straightforward introduction of new rules.

Table 2.1: Comparison of software development models

Model	**Strengths**	**Weaknesses**	**Most appropriate**	**Least appropriate**
Waterfall	Tight control; measurable development progress; extensive documentation; conserves resources.	Inflexible, slow, costly and cumbersome due to significant structure and tight controls.	Large, expensive and complicated projects with clear objectives and solution and well defined and stable requirements. Supports less experienced teams and managers, or project teams with unstable composition.	Large projects with high level of change or uncertainty. Web Information Systems (WIS), real-time or event-driven systems, leading-edge applications.

Table 2.1: (continued)

Model	**Strengths**	**Weaknesses**	**Most appropriate**	**Least appropriate**
Prototyping	Encourages innovation and flexible designs. Early deliverables	Can lead to poorly designed systems. Can lead to false expectations. Added costs of iterations	High-risk portions of very large, complex projects. Resolving unclear objectives; developing and validating user requirements; experimenting with design; investigating performance and the human computer interface.	Projects with unstable team composition, clear objectives and low project risk regarding requirements definition. Future scalability of design is critical.
Incremental	Mitigation of integration and architectural risks early in the project. Early deliverables and gradual implementation.	Difficult problems tend to be pushed to the future to demonstrate early success. Requires well-defined interfaces due to the different time of module releases.	Large projects where requirements are not well understood or are changing due to external changes, changing expectations, budget changes or rapidly changing technology. WIS and event-driven systems, leading-edge applications.	Very small and short projects with low integration and architectural risks. Highly interactive applications where the project largely comprises analysis or reporting of already existing data.
Spiral	Enhances risk avoidance. Can incorporate Waterfall, Prototyping, and Incremental methodologies depending on the type of project risk.	Highly customised to each project; limiting reusability; quite complex. No controls for moving from cycle to cycle; no firm deadlines. A risk of not meeting budget or schedule.	Projects with highly prioritised risk avoidance, requiring high accuracy, strong approval and documentation control. Real-time or safety-critical systems. Project manager is highly skilled and experienced. Implementation has priority over functionality, which can be added in later versions.	Minimising resource consumption is an absolute priority. Functionality has priority over implementation. A high degree of accuracy is not essential. Risk avoidance is a low priority.

Table 2.1: (continued)

Model	Strengths	Weaknesses	Most appropriate	Least appropriate
RAD	Fast development and delivery of high quality software at a relatively low cost. Tighter fit between user requirements and system specifications. Provides the ability to rapidly change system design.	More speed and lower cost may lead to lower overall system quality. Potential for inconsistent designs and lacking scalability. Difficult problems tend to be pushed to the future. Well-defined module interfaces are required. High cost of commitment on the part of key user personnel.	Short and small-to-medium scale projects ($\leqslant$ 6 man-years) with focused project scope and well defined and narrow business objectives. Application is highly interactive, has a clearly defined user group, and is not computationally complex. Requirements of the system are unknown or uncertain. Technical architecture is clearly defined. Team composition is stable. Effective project control is definitely available.	Very large distributed information systems, computationally complex applications, real-time or safety-critical systems. Projects with obscure business objectives, unclear technical architecture, large or multiple project teams and lacking user commitment.

Most of the numerous software development methodologies, which are used to direct real software projects (see section 1.1, figure 2.10), are actually build on "customised adaptations of the generic models", affected overtime by changing needs of computer customers and hardware and software evolution [6]. Significant variations between the real circumstances and the conditions the model was designed for, can lead to unpredictable results from applying the model to practice. Therefore it is sometimes necessary to alter the existing model or use a combination of models to accommodate the custom circumstances [6]. System development process models in use today are based on linear (Waterfall), iterative (Prototyping, RAD, Extreme Programming) or combined development (Incremental, Spiral) [10]:

A single software development project can combine two or more process models to mitigate risks, associated with applying one or another approach. For example, a development framework, based on spiral development can incorporate prototyping for requirements iteration in a project with high risk of not meeting user requirements and low risk of exceeding budget and time. In the opposite situation it would be beneficial to employ a waterfall model for the particular software life cycle iteration [10]. Similarly, RAD and incremental methodologies are sometimes used in conjunction with prototyping for managing high-risk portions of large and complex projects.

Previous research [88] attempted to derive major positive and negative past software engineering experiences (summarised in Table 2.2). The formulated timeless software development principles could help in selecting the most suitable approach for a particular software development project and lead to its successful implementation, while avoidance of the negative practices minimises project risks.

Table 2.2: Positive and negative software engineering experiences (based on [88])

Positive practices	**Negative practices**
• The software have to be useful to people. • Determined goal clearly envisioned by all stakeholders. • The scope of any software should be within the comprehension and development capacities. • Foreseeing possible outcomes, avoiding premature commitments. • Managing stakeholder expectations, satisfying value propositions of all stakeholders. • Early delivery of software at satisfactory quality, early return of investments. • Application of engineering approach for software development, which should involve mathematics, computer science, behavioural sciences, economics and management science, as well as use the scientific method to learn through experience. • Analysis is the key to early elimination and prevention of errors. • Creative thinking and prototyping frequently lead to novel and highly rewarded solutions at generally low risk. • Software reuse, prototyping, process improvement, tools, staffing, training could significantly improve productivity. • ”What is good for products is good for process, including architecture, reusability, composability, and adaptability”. • Adaptability is favoured over repeatability, when the rate of change is rapid.	• Using a rigorous sequential process usually result in slower development and is poorly suited for projects with high level of change and unpredictability. • Cowboy programming frequently leads to poor design and incorrect technology choices, which subsequently result in producing unused or unusable components before the project is finished. • Top-down development and reductionism often turn out to be unrealistic because of unclear requirements, indecisiveness of stakeholders, rapid changes, software reuse and off-the shelf solutions. • There are no universal solutions suitable for every software development project. • Unrealistic ambitious milestones may result in poor specifications and lots of rework. • Software developers should be cautious of overoptimistic expectations regarding the pace of technology capabilities maturing, as in the case with Model Driven Architecture (MDA).

In general, following a certain methodology when developing software enables better organisation and control over the process. Depending on the project scale, features and circumstances, which impact the development process, an appropriate model could be picked up and applied as it is or adjusted if necessary.

2.5 Research gap

Based on the research in applying the Web-based technology for the development of CAD/CAM systems, reviewed in this chapter, two main directions for further research are outlined. Firstly, an efficient way of exploiting state-of-the-art Web technologies in the area of CAD/CAM needs to be explored and formalised by identifying features specific to Web-based CAD/CAM software development and by comparing those against known software development methodologies. Introducing a dedicated methodology for Web-based CAD/CAM development will improve quality and reliability of Web-based CAD/CAM systems, enable their use in production and encourage developing more CAD/CAM systems using Web technology. The established methodology for creating Web-based CAD/CAM systems will then require application and validation.

Secondly, design and development of browser-based CAD/CAM systems has been little discussed within the literature and therefore is an important gap in the software engineering area. Building industrial CAD/CAM systems for Web still remains a challenging task, due to issues such as Internet connectivity and security, interactivity and graphics visualisation, sharing computations between the server and the client and user collaboration. At the same time, browser-based CAD/CAM systems allow better utilisation of Web technology advantages and could benefit many areas of industry.

Apart from that, considerable empirical evidence is desirable to demonstrate, that a Web-based approach provides significant advantages for the area of CAD/CAM software compared to similar desktop applications.

2.6 Summary

In this chapter, accumulated knowledge and experience in the areas relevant to the Web-based CAD/CAM software development are studied and the frontier of the current developments is explored. The analysis of publications about Web-based systems indicates increasing interest about this kind of software and in particular about Web-based PDM and DSS systems. The experience of applying Web technology for the development of these kind of systems covers many subject areas, with computer science and engineering being the largest categories. Software development trends show the increasing role of Web technology in software industry, which is governed by Web-based approach advantages.

Based on the analysis of technologies used for creating Web-based CAD/CAM systems, reviewed in the literature, these systems could be classified into two major groups: systems that are built with Web and systems that are built for Web. The latter appear the main interest for this research, as the development of these kind of applications is associated with greater challenges and provide better possibilities for utilising Web technology advantages. The main challenges for creating Web-based CAD/CAM systems are related with internet connectivity and security, requirements for interactivity, graphics visualisation, user collaboration, as well as creating optimal software design and selecting appropriate development methodology.

The analysis of Web-based CAD/CAM implementation methods and techniques shows, that many developers use integrated solutions and well known programming technologies, which allow minimising risks associated with novelty and complexity of the task. Integration with proprietary CAD components limits the extent of utilising Web-based technology advantages in terms of flexibility, compatibility and ease of maintenance.

Despite the plethora of Web-based technologies, there is no defined generic approach for the development of Web-based CAD/CAM systems. Almost half from all reviewed Web-based CAD/CAM software appeared to be prototypes developed as proof-of-concept applications. The development methodology and the design of Web-based CAD/CAM applications are usually based on trial and error.

A review of software development methodologies is performed to find out how a development methodology is selected based on the circumstances of each specific case, what are the strengths and weaknesses of existing methodologies, the positive and negative software engineering experiences, what are the possible variations in software development process model and how these models could be adapted and combined to enable more effective approach for software development in given circumstances. The approaches used for development of Web applications, CAD/CAM software, scientific CAD/CAE and complex CSE software are especially relevant for this research and current practices are reviewed in detail.

To summarise, the overview of software development trends, role of Web technology in industry and analysis of the previous experience in Web-based CAD/CAM software development confirmed the ability of WWW to serve as base technology for the development of complex industrial software systems and in particular CAD/CAM applications. However, the absence of a common approach for design and development of Web-based

CAD/CAM systems is associated with increased risks and challenges. To overcome them and facilitate successful implementation of this kind of software within the commercial environment, a systematic approach is required that will ensure robustness, effectiveness, usability and ease of maintenance of these applications. Thus, introducing a methodology for the development of Web-based CAD/CAM software is essential for enabling Web technology benefits in industry.

Chapter 3

Research aim, objectives and methodology

3.1 Research aim and objectives

The aim of this research is to develop a methodology for creating Web-based CAD/CAM software systems. In accordance with the terminology provided in Section 1.1, the new methodology should establish an approach to creating Web-based CAD/CAM software, assuming a specific collection of principles and practices, with a defined set and timing of software development activities, feedback and control methods, giving an advice on how to manage the quality of deliverables.

Several objectives are defined guide the research, namely:

1. Provide the rationale and prerequisites for the new software development methodology;
2. Establish a methodology for Web-based CAD/CAM system development;
3. Apply the proposed methodology on a set of CAD/CAM software development case studies;
4. While following the proposed methodology utilise Web technology advantages by developing industrial browser-based CAD/CAM software case studies;
5. Perform validation of the proposed methodology by evaluating the software developed using this methodology.

Fulfilling these objectives sequentially ensures the logical flow of the research from providing the rationale and prerequisites for the new methodology to describing the methodology for Web-based CAD/CAM systems and finally to applying it on practice, validating and evaluating.

3.2 Research methodology

The research methodology describes the way to accomplish the research aim and objectives. As shown in Figure 3.1, in accordance with the methodology the research process follows through the four stages: Analysis, Development, Application, Validation and evaluation. Every stage delivers the results of related research part and all together they comprise the basis of the research thesis.

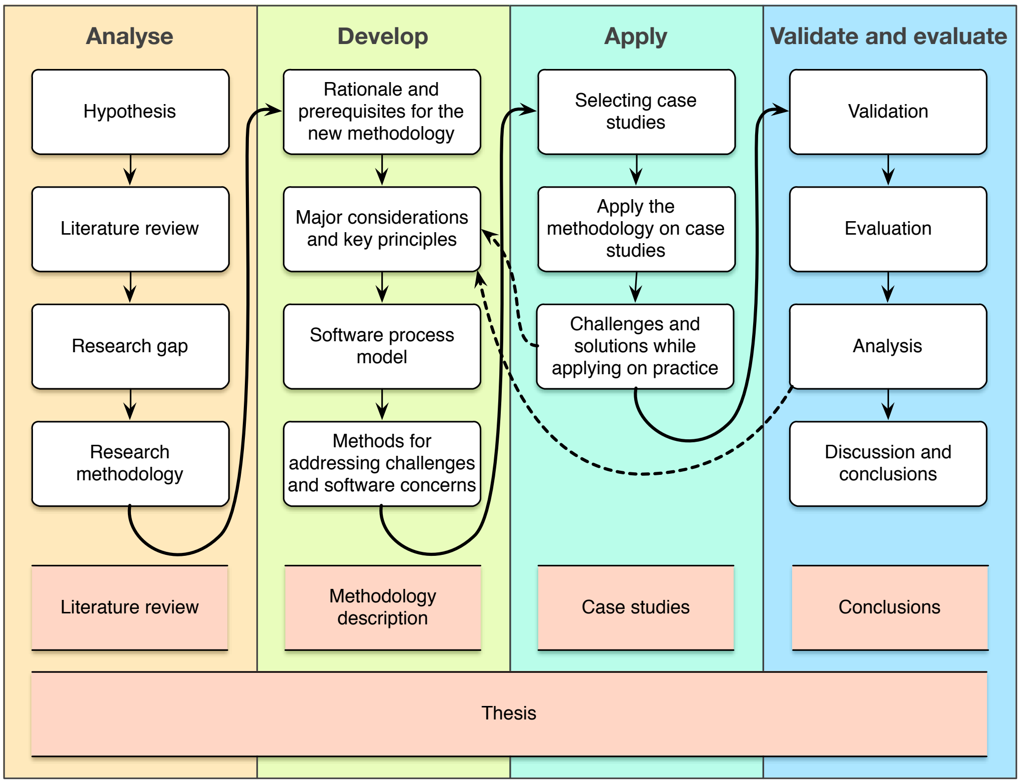


Figure 3.1: Conceptual diagram of the research methodology

3.2.1 Analysis

As a starting point for this research serves a hypothesis that Web-based approach provides significant advantages for CAD/CAM software development. The researcher then becomes familiar with the subject area, identifies and states the problem to be solved in this research. Accumulated knowledge and experience in the relevant areas are studied to explore the frontier of the current developments. For this reason the role of Web technologies in industry and common trends in software development are overviewed, paying considerable attention to the application of Web technology in product design and previous experiences in Web-based CAD/CAM software implementation methods and techniques. The research gap is then identified to set the focus for the further research. An appropriate research methodology is selected to direct research process providing guidelines for achieving the research aim and objectives. Steps, methods and tools used in the analysis stage of the research are presented in Figure 3.2.

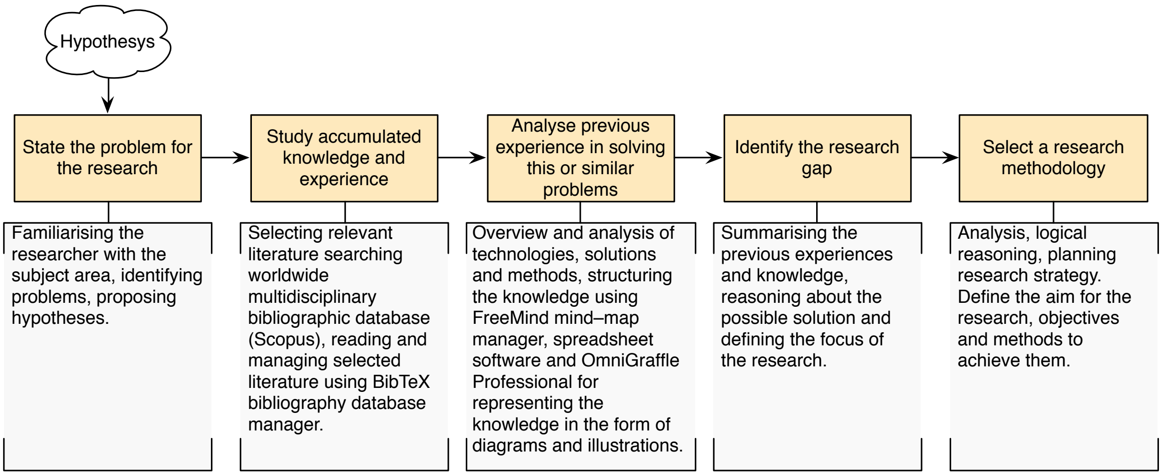


Figure 3.2: Steps, methods and tools used in the analysis stage

Figure 3.3 provides a flow chart of main steps and decisions guiding the research process in the analysis stage. First of all, to study the problem stated in Section 1.5 it is required to confirm the maturity of Web technologies and the ability of WWW to serve as base technology for CAD/CAM software. The aim of this step is to identify capabilities of utilising Web technology in the area of CAD/CAM or identify the obstacles for creating Web-based CAD/CAM software and directions for further improvement of Web technologies allowing to overcome these obstacles.

Then reasons for increasing popularity of Web applications are investigated and advantages and challenges of Web-based CAD/CAM software development are reviewed to define the expediency of creating this kind of applications. Further, the study reviews how Web technology is used in product design to find out any existing common approaches for utilising Web advantages in this area. In case of absence of ready solutions or common practices for Web-based CAD/CAM software the review of associated challenges can guide further investigation into the design and development of Web-based CAD/CAM software.

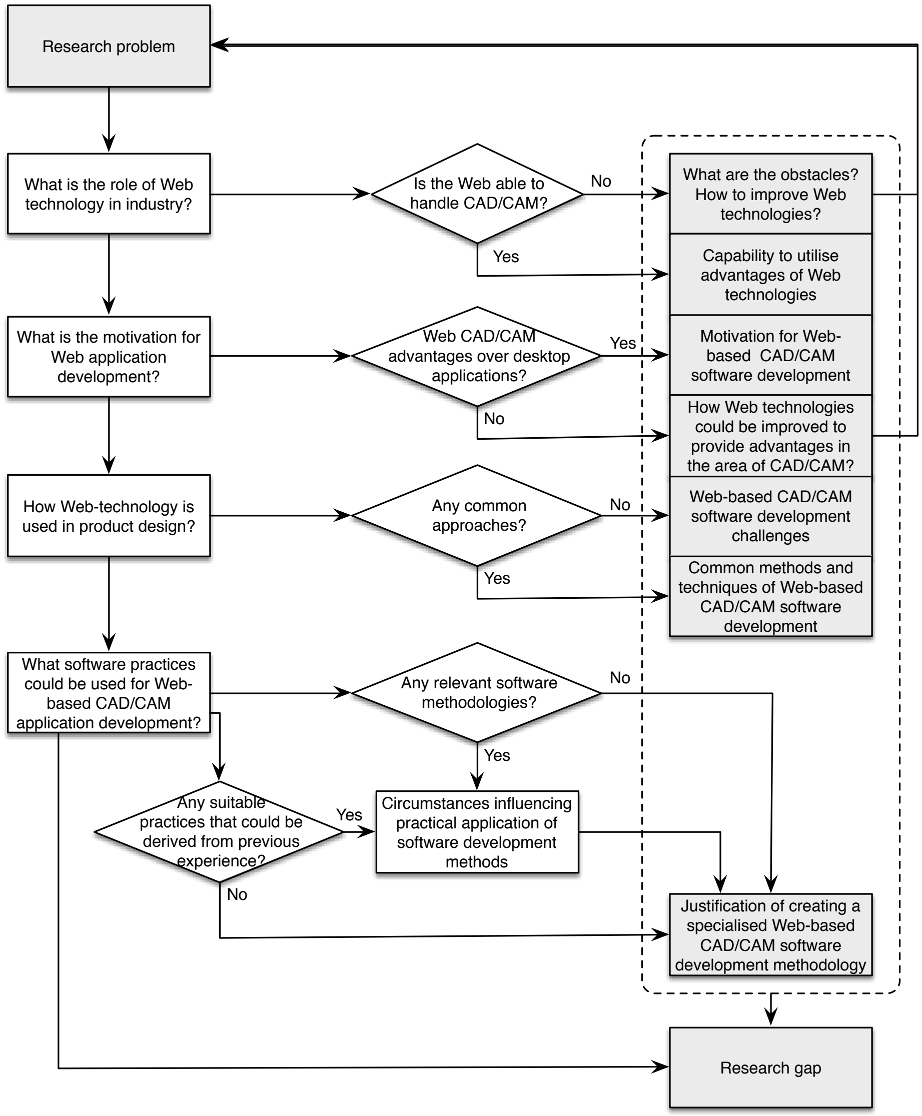


Figure 3.3: Flow chart of the research process in the analysis stage

After all, software methodologies relevant for the research are reviewed to find out if any of them could be applied for Web-based CAD/CAM application development. Circumstances influencing practical application of software methodologies are investigated to understand what factors facilitate effective control over the development process. Considering all the above reasoning for creating a dedicated Web-based CAD/CAM software development methodology is provided in the research gap.

3.2.2 Development

In this part of the research methodology for creating Web-based CAD/CAM systems is developed. This task is completed in four steps that are presented in Figure 3.4 along with the selected research methods and tools.

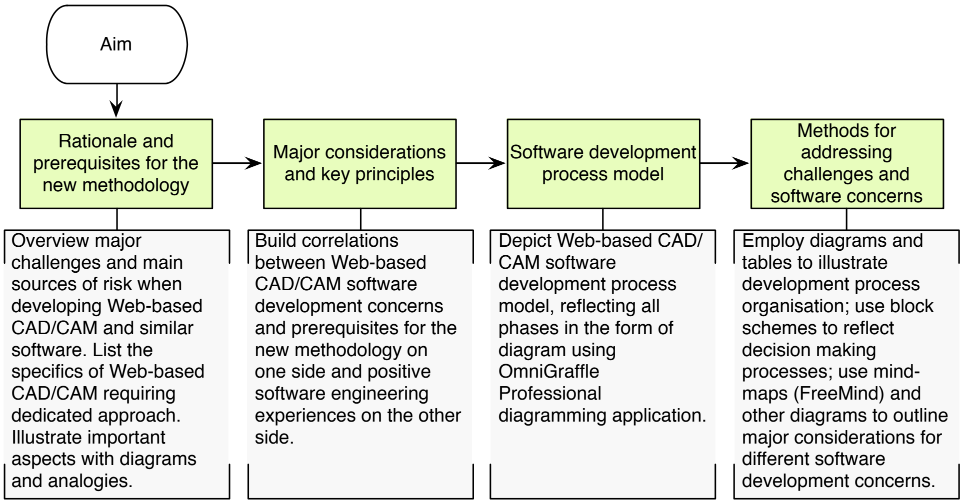


Figure 3.4: Steps, methods and tools used in the development stage

The first step provides rationale and prerequisites for the new methodology based on the overview of major challenges and main sources of risk associated with Web-based CAD/CAM software development. Then major considerations and key principles of the new methodology are constructed from the correlations between Web-based CAD/CAM software development concerns and prerequisites for the new methodology on one side and positive software engineering experiences on the other side. Web-based CAD/CAM software development process model is presented to provide guidance and control over the development and ensure the desired process performance. Finally, to establish more considerable guidance the methodology provides description of methods for addressing associated challenges and software development concerns.

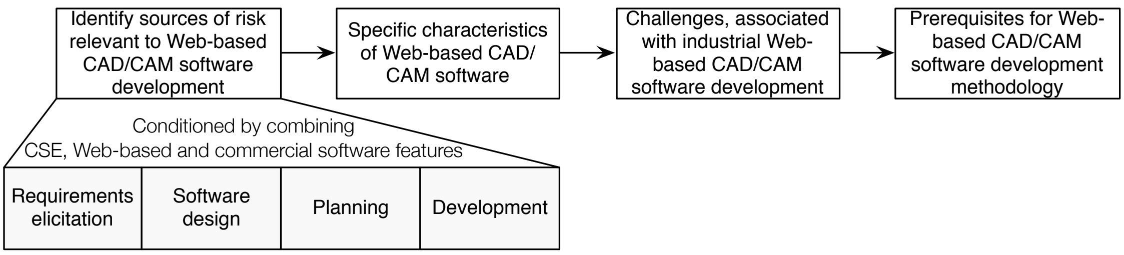


Figure 3.5: Providing rationale and prerequisites for Web-based CAD/CAM development methodology

As stated above, rationale and prerequisites for Web-based CAD/CAM development methodology are provided based on the overview of major challenges and main sources of risk associated with Web-based CAD/CAM software development. These challenges are conditioned by combining CSE, Web-based and commercial software features, and cover requirements elicitation process, software design, planning and development. These aspects of Web-based CAD/CAM software development are reviewed closely based on available literature and are focused on identification of the unique features of Web-based CAD/CAM software development and challenges, which need to be addressed in a special way. Based on these characteristics specific to Web-based CAD/CAM software challenges associated with its development are defined and prerequisites for the dedicated software development methodology are formulated. The steps undertaken to provide rationale and prerequisites for Web-based CAD/CAM development methodology are shown in Figure 3.5.

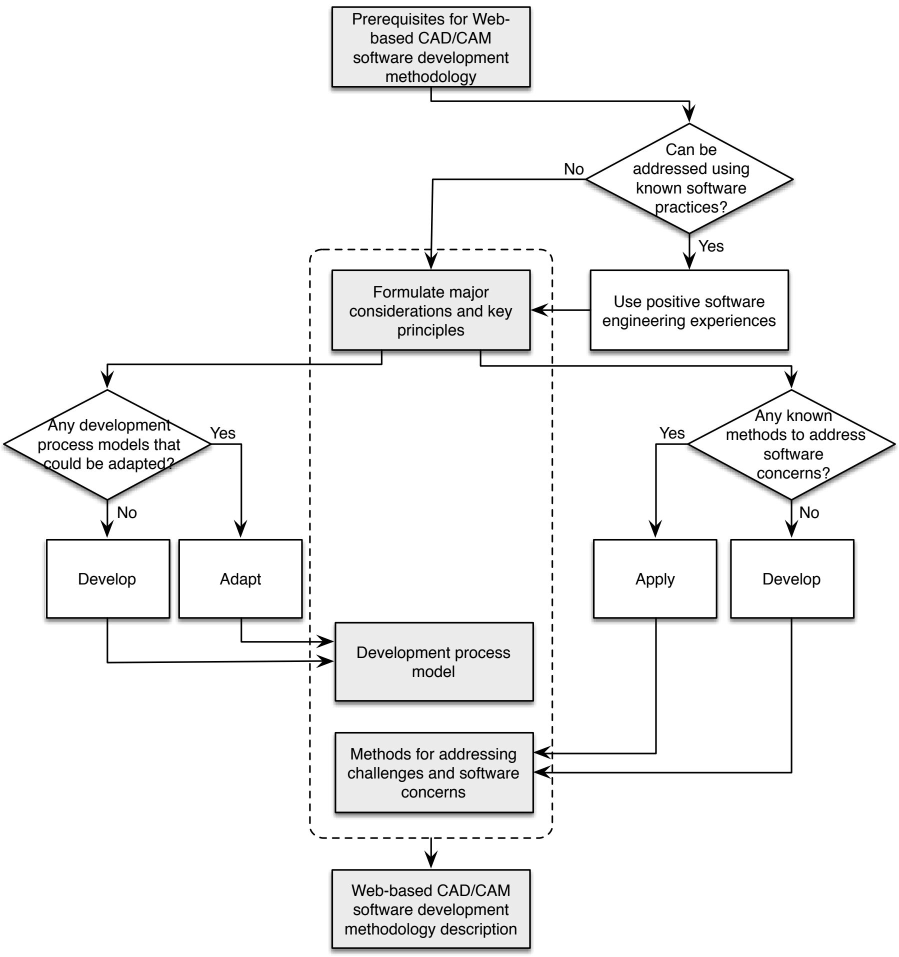


Figure 3.6: Steps and decisions used to construct a methodology for Web-based CAD/CAM development

The methodology for Web-based CAD/CAM software is constructed based on the formulated prerequisites by following through the steps and decisions presented in Figure 3.6. Major considerations and key principles for the new methodology are derived from the features specific to Web-based CAD/CAM software considering the positive software engineering experiences reviewed in Section 2.4.3. In the absence of appropriate ready to use software development process model it could be developed based on models known to be effective for addressing challenges relevant to Web-based CAD/CAM system development. Finally, methods for addressing particular software development concerns are selected from common software development practices or proposed based on the formulated key principles and considerations.

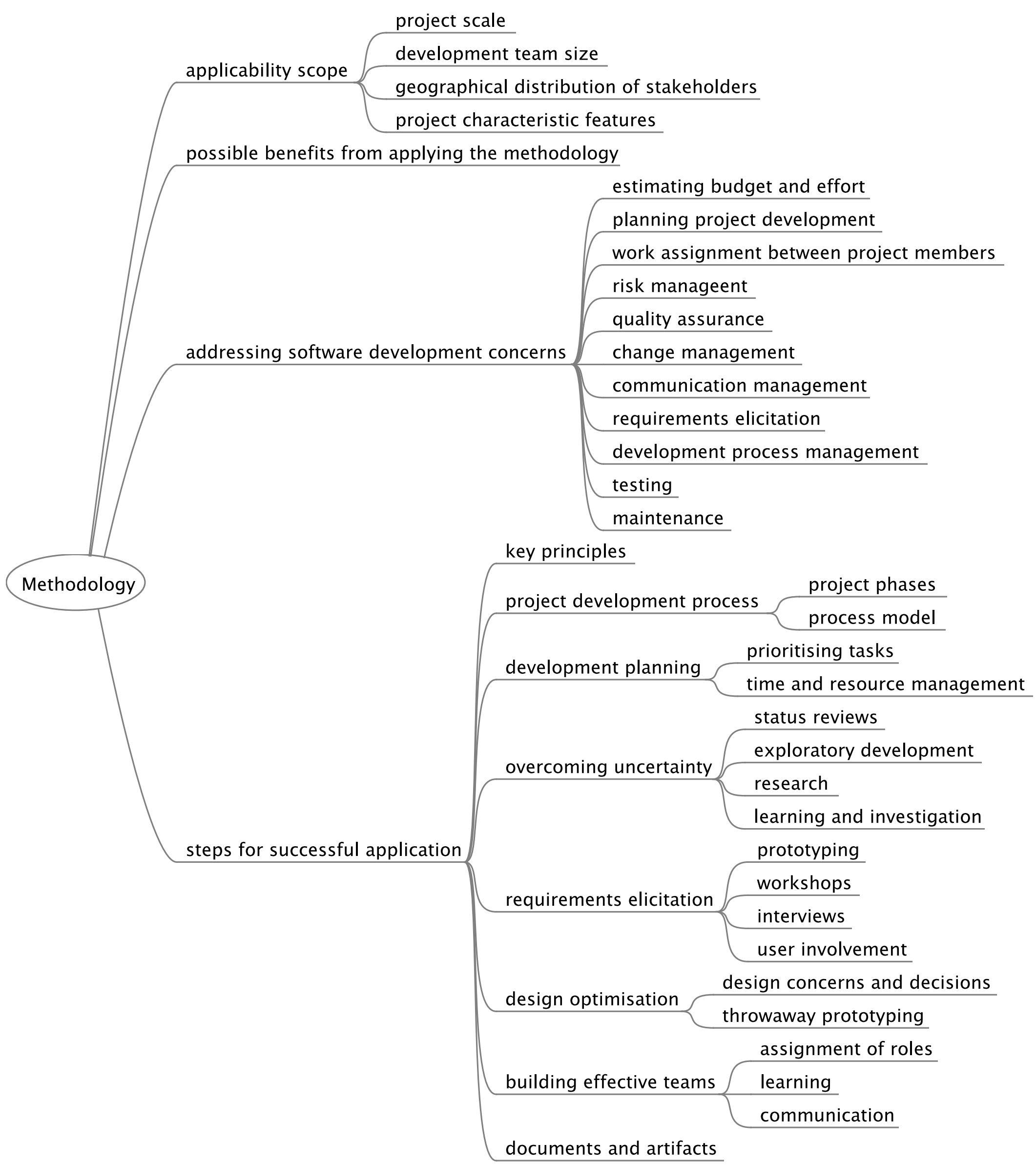


Figure 3.7: Concerns for Web-based CAD/CAM software development methodology

When describing a methodology for developing software products a number of concerns are kept in mind. Those are mind-mapped in Figure 3.7 and include the scope of applicability for the proposed methodology, possible benefits from its application, how the typical software development concerns are addressed and steps that are necessary to undertake for the successful application of the methodology. These methodological concerns are used in the research to develop the methodology for creating Web-based CAD/CAM software.

3.2.3 Application

In the application stage the proposed methodology for Web-based CAD/CAM software development is generalised on practice to confirm its applicability and explore the challenges that may be faced by the developers while following this methodology. Steps, research methods and tools used in this part of the research are presented in Figure 3.8, while Figure 3.9 provides the flowchart of the research process in the application stage.

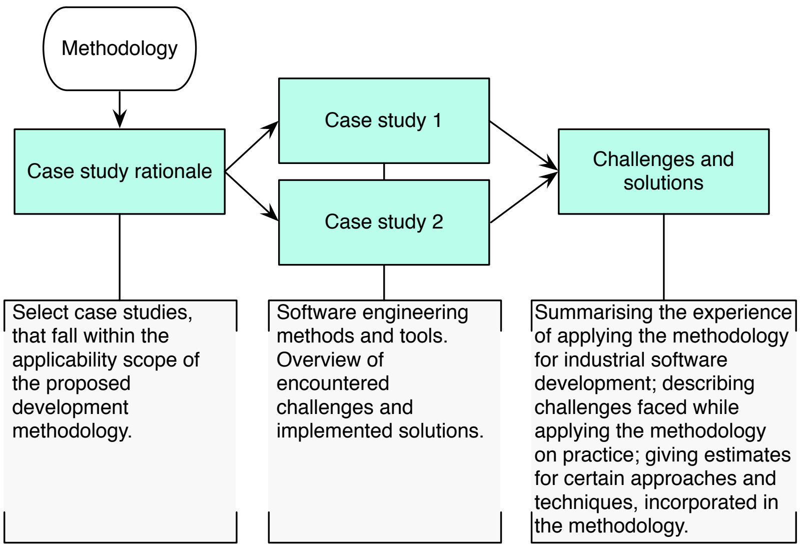


Figure 3.8: Steps, methods and tools used in the application stage

The best way to confirm the effectiveness of the methodology for creating Web-based CAD/CAM systems is development of real case studies while following this methodology. As this research is funded by the KTP program, a range of case studies has been chosen within the sponsor company. To ensure generality of research results selected case studies deal with different things, namely, one case study is focused mainly on design process automation and the other case study solves tasks associated with computer-aided manufacturing. Challenges faced during the application of the methodology for the case study

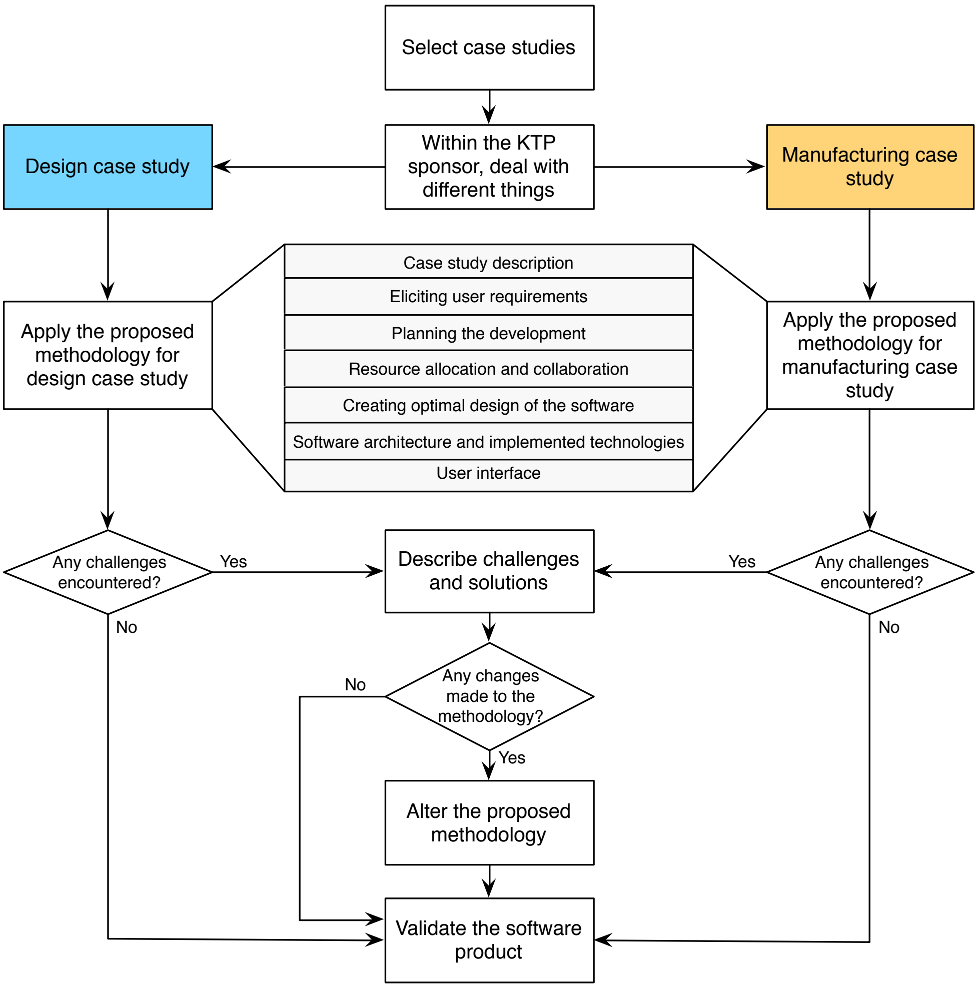


Figure 3.9: Flowchart of the research process in the application stage

development are discussed at the end of this chapter and the alterations to the proposed Web-based methodology are given if any are necessary.

Both case studies are developed by the same team of developers, this way ensuring equal circumstances from the developers side, such as their skills, knowledge and expertise. Although the case studies are developed sequentially one after another, this should not have any significant impact on the success of the second case study, as the chosen case studies are related with different areas of expertise. To stress test the methodology and explore the challenges, associated with its application in real life, the resulting Web-based CAD/CAM systems have to meet the following conditions:

- ensure all functional and non-functional requirements and be ready for use in production,
- accomplish all business processes associated with application area,

- ensure robustness, performance, reliability, security and usability comparable with the standards of modern commercial software,
- utilise advantages of Web-based applications to the highest extent.

3.2.4 Validation and evaluation

To prove the effectiveness of the proposed methodology for Web-based CAD/CAM software development its validation and evaluation is performed through the validation of case study results, evaluation of the developed CAD/CAM applications using specialist opinions and analysis of advantages and limitations of these applications. The results of this part of the research are then summarised and conclusions about the proposed methodology are made. Figure 3.10 describes research methods for each step in the validation stage.

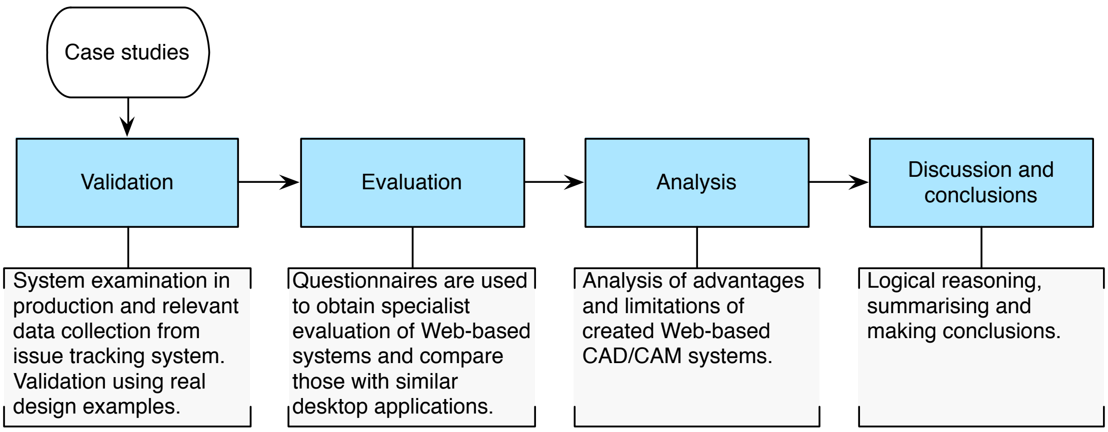


Figure 3.10: Steps, methods and tools used in the validation and evaluation stage

Figure 3.11 presents the flowchart illustrating the validation process and conclusions facilitated by each section in the validation stage. Particularly real design examples are used to confirm validity of results and software compliance with requirements and support defined business processes. Readiness for use in production and compliance with business requirements is further confirmed through industrial use of the developed Web-based CAD/CAM software products. Specialist opinions are used to estimate benefits from using Web-based CAD/CAM systems over similar desktop applications. Finally, advantages and limitations of the developed Web-based CAD/CAM software are analysed to summarise strengths and weaknesses of the software solutions, implemented by following the proposed software development methodology. By summarising the outcomes of validation, evaluation and analysis of the developed software products a conclusion about the effectiveness of the proposed methodology for creating Web-based CAD/CAM software is made.

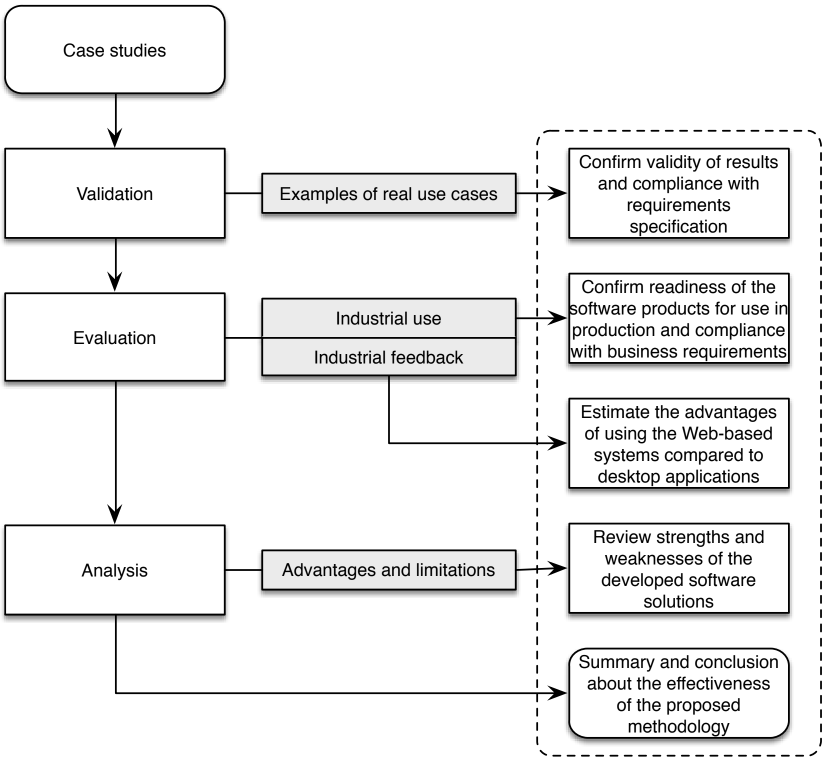


Figure 3.11: Method for validating the proposed methodology for Web-based CAD/CAM development

3.3 Summary

This chapter defines the focus of the research and describes the way for achieving its aim and objectives. The target of this research is to develop a methodology for creating Web-based CAD/CAM software systems, and the objectives ensure the logical flow of the research from providing rationale and prerequisites for the new methodology to its description, practical application, validation and evaluation.

The largest part of this chapter is devoted to the description of the research methodology, that presents a four stage research process, incorporating analysis, development, application, as well as validation and evaluation. The research methodology provides a clear guidance on what is required to be done in every stage of this research, describes methods and tools for accomplishing these requirements and states the deliverables expected from every stage execution.

The research strives to cover all identified knowledge gaps, namely:

- delivering a dedicated methodology for Web-based CAD/CAM development,
- focusing on discussion of browser-based CAD/CAM system design and development,
- demonstration of Web-based CAD/CAM software advantages compared to similar desktop applications.

Chapter 4

Rationale and prerequisites for Web-based CAD/CAM development methodology

To provide rationale and prerequisites for Web-based CAD/CAM development methodology, this Chapter identifies features specific to Web-based CAD/CAM software and associated challenges, which need to be addressed in a special way. The review of challenges and sources of risk in Web-based CAD/CAM software development covers requirements elicitation process, software design, planning and development. Finally, the considerations for creating specialised methodology for Web-based CAD/CAM software are provided based on the identified features of industrial Web-based CAD/CAM software and challenges associated with its development.

4.1 Science-intensive software

Industrial CAD/CAM system development could be viewed as development of a system, that combines features of both CSE and commercial software. It is important to note, that development methodologies for CSE software and commercial applications are different [103]. While the comprehensibility, maintainability and extensibility are regarded as essential in commercial software development, the major importance of CSE application development are the correctness and efficiency of the software [103]. Thus, a Web-based CAD/CAM system would need to combine usability, scalability, maintainability and possibilities for collaboration inherent to Web applications with the reliability, complexity, interactivity and computational performance of CAD/CAM and CSE software.

Creating Web applications differs from the development of industrial software with capabilities for complex computations and simulations. The latter usually presumes the necessity to conduct a number of small studies, creating and verifying prototypes of separate software modules during the project development [99]. "Adoption of formal software practices by CSE projects is hindered by the fact that the software product is primarily a vehicle for producing science and engineering results. In this kind of projects too much formality can do more harm than good" [104].

Combining CSE, Web-based and commercial software features while developing a single software product is a challenge due to the absence of a methodology able to address the associated set of challenges. Moreover, as shows the practice, even software development methodologies providing comprehensive guidance often cannot be applied without making adjustments for a particular project circumstances [99, 105].

4.2 Requirements elicitation

Requirements elicitation process in software development projects is usually coupled with scope, understanding or volatility problems [106]. The problems of understanding tend to naturally arise between people with different background and expertise areas, who's vocabulary and perception of real world may be very different. Scope problems are caused by either insufficient or overly detailed, confusing information, provided by stakeholders. And finally, user requirements can just change with time, making the requirement elicitation process even more challenging.

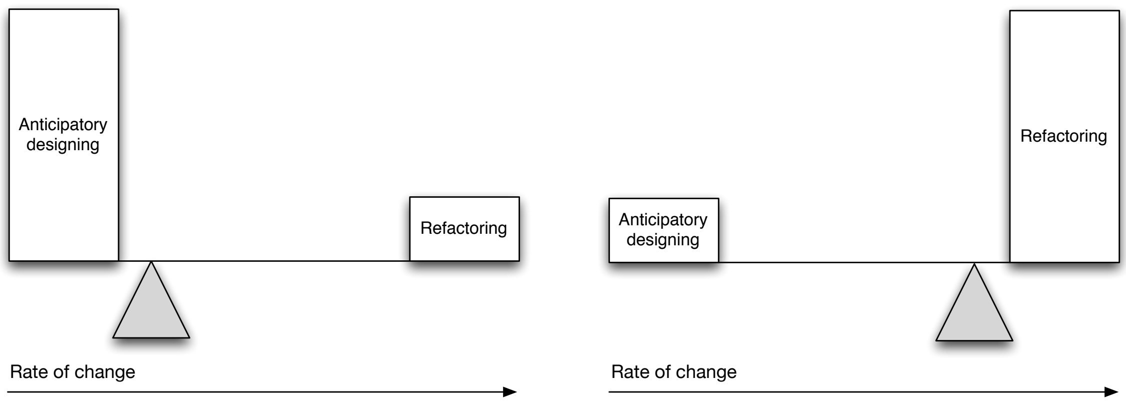
Exploratory development process is not seldom for complex CAD/CAM software system projects. Basic software requirements obtained in the beginning of the project are extended and refined later on during the development. As the development of this kind of software usually takes long time, it can last for several years, introduction of new features and technologies can face certain difficulties. New requirements and conditions often require significant changes in software architecture [99].

The high degree of specification ambiguity is relevant to CAD/CAM software development, mainly due to the diversity of expertise of the parties, involved in the process. Therefore, a flexible development model would be beneficial for such projects as it allows to cope with continuously changing requirements better.

4.3 Design

As it was noted in the previous section, CAD/CAM software development usually can be characterised as projects with high level of change, mainly because of the uncertain requirements and specification ambiguity. Rapidly changing business environment only adds its two cents to the level of change in the project. Software design adjustability and expandability both matter in CAD/CAM system development. Therefore, CAD/CAM software require a design, that can be easily modified at any stage of the development. Correct balance between the design, development and test time can minimise software errors and boost system robustness.

The problem of balancing between anticipatory design and refactoring has been discussed in the literature [107]. The lower is the rate of change, the more anticipatory design versus refactoring may be reasonable (as shown in Figure 4.1(a)). Subsequently, in the environments experiencing high rates of change the balance point between anticipatory designing and refactoring is shifted more towards refactoring (as shown in Figure 4.1(b)). Nevertheless, the quality of initial design remains extremely important.



(a) At lower rates of change

(b) At higher rates of change

Figure 4.1: Balancing design and refactoring [107]

It is hard to produce a perfect architectural design up front, because at the start of the project the knowledge about the software being developed is imperfect. Developers have tried different ways to approach this problem, for example, by deferring big decisions and concentrating on getting the little ones right, like in Test-Driven Development (TDD) [108, 109], or by attempting to identify significant problems earlier in the project, like in iterative development [110].

Upstreaming software design optimisation is essential for creating reliable software on time and budget. Adherents of Robust Software Development Model (RSDM) assert "that because errors in software are almost all created well upstream in the design process, and because software is all design and development, with no true manufacturing component, everything that can be done to create bug-free software must be done as far upstream in the design process as possible" [111].

Also design flexibility is especially important when developing computationally intensive software. The experience with procedurally implemented and speed-optimised algorithms [112] confirms the exponentially increasing cost of changing the fundamental design of elaborate code. Choosing technologies with some reserve in performance, flexible design and high-quality structured code should help to avoid premature code optimisation.

4.4 Planning

In contrast to traditional 'waterfall' development, a complete up front design cannot be achieved in projects with uncertain requirements. Hence creating a predictable schedule and budget for the development with evolutionary design in mind is a big challenge. In the case with evolutionary design the design is essentially a part of the programming process and as the program evolves the design changes. To some extent it is a creative process, which by definition is not easy to plan.

Based on software design and development expertise one can conclude, that unlike civil engineering software construction costs are incomparably cheap, while design makes the most effort [113]. This makes agile development justifiable for the projects with uncertain or changing requirements.

One of the most important prerequisites for dealing with unpredictability successfully is accurate knowledge about current situation [113]. This could be achieved through frequent feedback mechanism lined up with iterative development [92].

Although, the flexibility of incremental model allows responding to specification changes and tackling smaller parts of complexity, its implementation for the projects incorporating research and development tasks is associated with the risk that learning can overweight the actual development lead to exceeding planned time and budget [111].

Thus, there is a need for a mechanism enabling adjustments in planned effort considering the results of research activities and exploratory development, while still staying focused on project objectives.

4.5 Incremental development challenges

As it was remarked previously, the incremental development provides a flexible model for response to changes. Unfortunately, incremental model has some obvious drawbacks inherent to many other models. As software project develops, it takes increasingly more effort to change something, that has been introduced far upstream in the project.

To illustrate this problem an analogy could be found in classic mechanics. Figure 4.2 shows software build-up in the incremental development. The rod illustrates the software product, which size grows with every iteration. Let's assume that the length L of the rod is the number of implemented function points and m is the size of the developed portion of the software provided in the number of Lines Of Code (LOC).

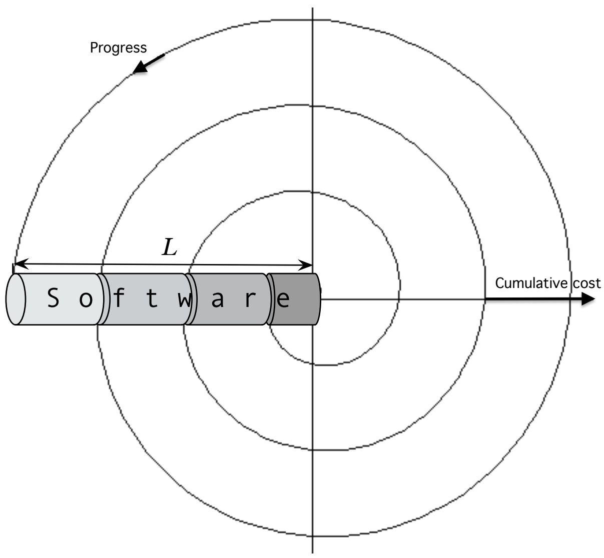


Figure 4.2: Software build-up in incremental development

In classical mechanics, "moment of inertia is a measure of an object's resistance to changes to its rotation" [114]. Increasing the mass increases the moment of inertia, and "distributing the mass further from the centre of rotation increases the moment of inertia by a greater degree" [114].

The moment of inertia I of the rod with the axis of rotation at the end of the rod can be calculated using the following formula:

$$I = \frac{mL^2}{3} [114],$$

where m is the mass of the rod in kilograms and L is its length in metres.

Subsequently, the work T required to accelerate a rod of a given mass from rest to its stated angular velocity ω expressed in radians per second, can be calculated using the formula:

$$T = \frac{\omega^2 I}{2}[114].$$

Considering that a rigid body consists of N point masses m_i whose distances to the axis of rotation are denoted r_i, the total kinetic energy T (or the work needed to accelerate it from rest to stated angular velocity) translates into [114]:

$$T = \sum_{i=1}^{N} \frac{1}{2} m_i v_i^2 = \sum_{i=1}^{N} \frac{1}{2} m_i (\omega r_i)^2 = \frac{1}{2} \omega^2 \Big(\textstyle\sum_{i=1}^{N} m_i r_i^2 \Big)[114].$$

In turn, the moment of inertia will be equal to:

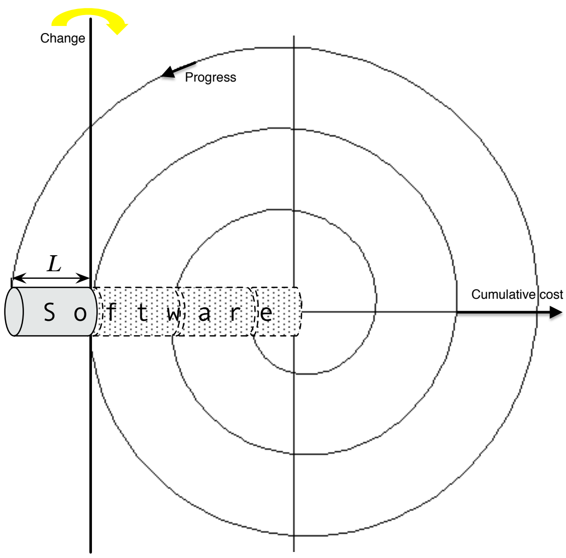
$$I = \sum_{i=1}^{N} m_i r_i^2[114].$$

Similarly to the work in classical mechanics, the effort required to make a change in software design is directly proportional to the moment of inertia of the developed portion of the software. In turn, the more is the length of the project (the number of implemented function points), the greater will be the moment of inertia (e.g. the resistance to the change).

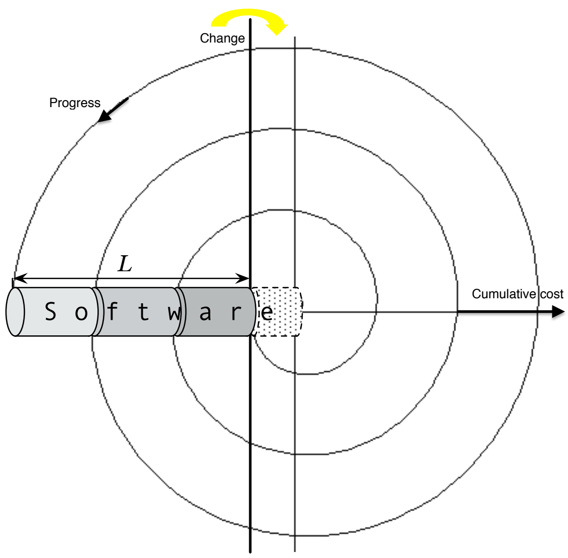
Figure 4.3 demonstrates two cases: when requested change affects a feature, that has been introduced recently, and in the other case, when it affects something, that has been introduced well upstream the development process.

To illustrate the moment of inertia in both cases, a new axis of rotation has been drawn through the point, where the change is required. The length of the acquired segment in Figure 4.3(a) is notably smaller than in Figure 4.3(b). Hence, in the second case the moment of inertia will be greater and it will take more effort to implement the change.

Software developers used to approach the problem of the increased moment of inertia, when making changes to older parts of the program, differently. Refactoring being one of them, helps to partially improve the design of the software by reviewing and redesigning those parts of the system, that are affected by the current development or required



(a) Small distance



(b) Long distance

Figure 4.3: Moment of inertia at different distances between the current state and change

changes. The idea of solving this problem is commonly based on the desire to minimise the necessity to make critical changes downstream the development.

4.6 Features specific to Web-based CAD/CAM software

To be able to approach the development of Web-based CAD/CAM software effectively a clear understanding of associated challenges is required. Application of one of the existing methodologies is troublesome due to the characteristics specific to Web-based CAD/CAM software, such as:

1. Requirements containing specific industry-related and often scientifically-intensive information;
2. Ambiguity of requirements and high level of change;
3. High level of uncertainty;
4. Fast pace of technology change;
5. Development process stretched over time;
6. Continuous application evolution;

7. Working team usually is not a group of experienced developers, but rather is created specifically for developing a particular software;

4.7 Web-based CAD/CAM software development challenges

The features to Web-based CAD/CAM software subsequently lead to a set of challenges that need be tackled when developing industrial Web-based CAD/CAM applications in a time and cost effective manner. The challenges include:

1. Scope, understanding and volatility problems during software requirements elicitation process.

2. Creating a predictable project schedule and budget.

3. Very difficult to produce a complete up front design due to high level of uncertainty.

4. Software architecture has great impact on overall scalability, performance and maintainability of CAD/CAM system.

5. Introduction of new features and technologies is complicated requiring significant architectural changes.

6. A need may arise for theoretical research and in depth problem investigation by field experts.

4.8 Prerequisites for specialised methodology

Taking into account the features specific to Web-based CAD/CAM software and challenges associated with the development of this kind of software, prerequisites for creating specialised methodology are identified in Figure 4.4. As it follows from the diagram, Web-based CAD/CAM software development process should consider possibilities for:

1. Gradual approach to coping with system complexity.

2. Addressing unpredictability in the development process.

3. Avoiding or minimising the necessity to make critical software design changes late in development.

Figure 4.4: Deriving prerequisites for the specialised methodology from features specific to Web-based CAD/CAM software and development challenges

4. Creating flexible design able to support further development of the CAD/CAM system.

5. Balancing between anticipatory design and refactoring.

6. Balancing between research and development and formal methodology.

7. Conducting small studies, developing and verifying prototypes of software system or its parts.

8. An approach to planning enabling adjustments in planned effort considering the results of research activities and exploratory development.

9. Clear communication of research results to all parties involved in the relevant task implementation.

10. Maintaining project documentation, accumulation and communication. The project documentation should be adequate and up to date.

11. An approach to control the amount of research and development.

12. Emphasis on close customer partnerships throughout the project development.

13. Minimisation of misunderstandings and elimination of barriers in communication between stakeholders.

Thus, the main prerequisites for the Web-based CAD/CAM software development methodology are formulated. The methodology is expected to address challenges governed by features specific to Web-based CAD/CAM applications and should provide techniques that would enable effective and quality software development.

4.9 Summary

Rationale and prerequisites for creating a Web-based CAD/CAM development methodology are provided in this Chapter. For this reason features specific to Web-based CAD/CAM software are discussed and associated challenges, which need to be addressed in a special way, are reviewed.

A Web-based CAD/CAM system would need to combine usability, scalability, maintainability and possibilities for collaboration inherent to Web applications with the reliability, complexity, interactivity and computational performance of CAD/CAM and CSE software. Combining CSE, Web-based and commercial software features while developing a single software product is a challenge due to the issues with adoption of formal software development practices by CSE projects, as those usually presume the necessity to conduct a number of small studies, developing and verifying prototypes of separate software modules during the project development.

The challenges and sources of risk in Web-based CAD/CAM software development are associated with requirements elicitation process, software design, planning and development. The high degree of specification ambiguity is relevant to CAD/CAM software development, mainly due to the diversity of expertise of the parties, involved in the process. The prolonged development of complex software systems tend to cause difficulties in introduction of new features and technologies, usually requiring significant changes in software architecture. Scope, understanding and volatility problems are relevant for Web-based CAD/CAM software requirements elicitation.

Because of the challenging software requirements elicitation, design adjustability and expandability matter in CAD/CAM system development. Design flexibility is especially important when developing computationally intensive software due to the exponentially increasing cost of changing the fundamental design of elaborate code. Quality of initial design is extremely important for creating reliable software in a time- and cost-effective manner. All of this makes the problem of balancing between anticipatory design and refactoring critical for the success of Web-based CAD/CAM software development.

Planning the software development process without a complete up front design, which is the case of Web-based CAD/CAM system development, is a big challenge. Incremental development could be used to address unpredictability and complexity in Web-based CAD/CAM system development process, but it brings in a risk that learning can overweight the actual development lead to exceeding planned time and budget. To deal with this there is a need for a mechanism enabling adjustments in planned effort considering the results of research activities and exploratory development, while still staying focused on project objectives.

While the incremental development provides a flexible model for response to changes, it has another important drawback that matters for prolonged development of complex software with high level of requirement uncertainty. As software project develops, it takes increasingly more effort to change something, that has been introduced far upstream in the project. An analogy from classic mechanics is used to illustrate this problem with the moment of inertia as a measure of an objects resistance to changes. When requested change affects something, that has been introduced well upstream the development process, it takes more effort to implement the change, than in the case with changing a feature, that has been introduced recently. The idea of solving this problem of increased resistance to change when modifying older parts of software is commonly based on the desire to minimise the necessity to make critical changes downstream the development.

Application of one of the existing methodologies is troublesome due to the characteristics specific to Web-based CAD/CAM software discussed in this Chapter. Taking into account the features specific to industrial Web-based CAD/CAM software, which are listed in Section 4.6 and associated challenges listed in Section 4.7, Web-based CAD/CAM software development process should consider prerequisites formulated in Section 4.8.

Chapter 5

A methodology for creating Web-based CAD/CAM systems

This Chapter constructs the methodology for creating Web-based CAD/CAM software systems using the approach described in Section 3.2.2. At the beginning of the Chapter, major considerations are given to address the challenges specific to Web-based CAD/CAM system development and aid the development process. Based on these considerations, key principles are outlined that need to be followed when developing Web-based CAD/CAM systems; also software development process model is established. Special attention is paid to planning approach, strategies for overcoming uncertainty during the project development and Web-based CAD/CAM software design concerns. Finally, section 5.9 describes the scope of projects that could be successfully developed using the proposed methodology, and lists the situations that could cause difficulties for applying this methodology.

5.1 Major considerations and key principles

Web-based CAD/CAM system development is associated with a number of challenges, which were discussed in chapter 4 and summarised in section 4.9. Table 5.1 gives some major considerations for addressing the identified challenges and aiding the development process of Web-based CAD/CAM systems.

Table 5.1: Major considerations for Web-based CAD/CAM software development

Development concerns	**Considerations**
Gradual approach to coping with system complexity.	Big and complex tasks should be divided in smaller parts, that are easier to deal with. The development would benefit from iterative and incremental process implementation. Development prioritisation should take into account task complexity and relevance to current business situation, giving preference to most complex and critical tasks to be researched and developed first.
An effective approach to cope with unpredictability in the development process.	This could be achieved by reducing uncertainty, improving adaptation to the change and minimising the impact of the change on the project. A set of activities that could be performed in this regard include frequent reviews and feedback, research of state-of-the-art technologies and standards.
Avoiding or minimising the necessity to make critical software design changes downstream the development.	Using throwaway prototyping for architecture optimisation at the beginning of the project and further refactoring to ensure the quality of the software design. Investigation into creating flexible design that suits best for project needs.
Creating flexible design able to support further development of the CAD/CAM system.	Investigation into technologies, standards and best practices, estimating risks and potential for the development of the system during the project and beyond.
Balancing between anticipatory designing and refactoring.	Creating optimised software architecture based on specification of key requirements with further exploration of less critical requirements and refining of application design.
Balancing between research and development and formal methodology.	Allow for a possibility to deviate from initial schedule (imperfect due to requirements uncertainty), keeping the overall project progress under control. Schedule deviations would enable development process adaptations for better utilisation of research results.
Conducting small researches, developing and verifying prototypes of software system or its parts.	A research method and milestones should be agreed before solving each problem situation. Reviews should be held at the end of dedicated time-box to decide on further actions based on the research progress and results obtained to date.
An approach to planning enabling adjustments in planned effort considering the results of research activities and exploratory development.	Project planning could incorporate time reserves to address the unpredictability issues. Thus, the required effort estimates could be continuously adjusted and become more accurate as the project is developed.
Clear communication of research results to all parties involved in the relevant task implementation.	Communication between all involved parties should be organised in an efficient and effective way. Meetings, workshops, reports and project documents could be all used for this purpose.

Table 5.1: (continued)

Development concerns	**Considerations**
Maintaining project documentation, accumulation and communication. The project documentation should be adequate and up to date.	Every project team member is responsible to prepare and keep up to date project documentation relevant to their part of work. Project manager should accumulate all the documentation and ensure it is available for involved people.
An approach to control the amount of research and development productivity.	The most complex, critical and high risk tasks should be researched and implemented first, thus reducing overall uncertainty in the project as the development approaches the end. Implementation of complex tasks and research activities should incorporate regular progress reviews risk estimates.
Emphasis on close customer partnerships throughout the project development.	Requirements elicitation, participation in workshops, feedback, product acceptance, expectations management.
Minimisation of misunderstandings and elimination of barriers in communication between stakeholders.	Project plan should reserve time for staff learning and knowledge exchange between parties with different expertise backgrounds.

Based on the considerations for addressing Web-based CAD/CAM system development challenges the following key principles should be kept in mind during the development:

- User involvement is vital for accurate and timely feedback on the project development.
- The development process is iterative and incremental and focused on frequent delivery based on task prioritising.
- Development prioritisation takes into account task complexity and relevance to current business situation, giving preference to most complex and critical tasks to be developed first.
- Big and complex tasks should be broken down to smaller chunks, that are easier to deal with.
- Exploratory requirement identification should be applied to eliminate ambiguity and deal with requirement uncertainty.
- Throwaway prototyping, design patterns and refactoring are used to ensure the quality of the software design.

- Project planning incorporates time reserves to address the unpredictability issues. Required effort estimates are continuously adjusted and become more accurate as project is developed.
- Communication between all involved parties should be organised in an efficient and effective way.

5.2 Software development process

This section describes phases and process model for the development of Web-based CAD/CAM software. The proposed software life-cycle builds on features of other widely used models and extends them to address the methodology prerequisites formulated in Chapter 4.

5.2.1 Project phases

The methodology consists of the following sequential phases:

- *Initiation.* In this phase project is initiated and preliminary research into the domain is conducted with the aim to get insight into the business needs and associated processes, develop the concept of the software, as well as perform preliminarily project risk assessment.
- *Design optimisation.* The aim of this phase is creating a good initial design for the software system. Key architectural strategies are defined and tried out using throwaway prototypes, that include only some basic functionality, critical for the architectural decisions.
- *Development.* After the optimal architecture for the software is defined, the system is developed iteratively and incrementally. The development process may involve search for solutions as problems arise due to the high uncertainty in the project, thus require additional short-term research and development of small prototypes for the possible solutions.

5.2.2 Development process model

The project development process model proposed by the methodology is shown in Figure 5.1. The model illustrates the downward development of the software, beginning with the initiation of the project, which incorporates preliminary research into the domain, the development of the software concept, assessing risks associated with the project development and specifying initial requirements.

After the initial phase the software architecture is developed, it is the core difficult-to-change elements of the system are created to provide infrastructure for the further functional enhancements.

Despite the reduced amount of initial design done in agile projects (for the reference see section 4.3), its quality remains extremely important. Improved quality of the initial design allows eliminating the necessity of making critical changes in software design during the incremental development (the impact of the moment of inertia of critical design changes explained in section 4.5). The main concern for creating a good architectural design up front is insufficient knowledge about the software developed.

The methodology enables to investigate the optimal architecture for the application before getting to the actual incremental development. For that reason throwaway prototypes are first developed based on initial set of requirements. This approach gives the developer better understanding of the software developed and enables early exploration of critical design features.

After the optimal architecture for the future software is defined, the system is developed incrementally following these basic steps on each iteration:

- Refining requirements
- Extending design documentation
- Assessing risks
- Prioritise tasks
- Short-term planning and adjusting time and effort estimation
- Refactoring
- Implementation
- Testing and fixing discovered errors
- Integration with the production system
- Obtaining feedback

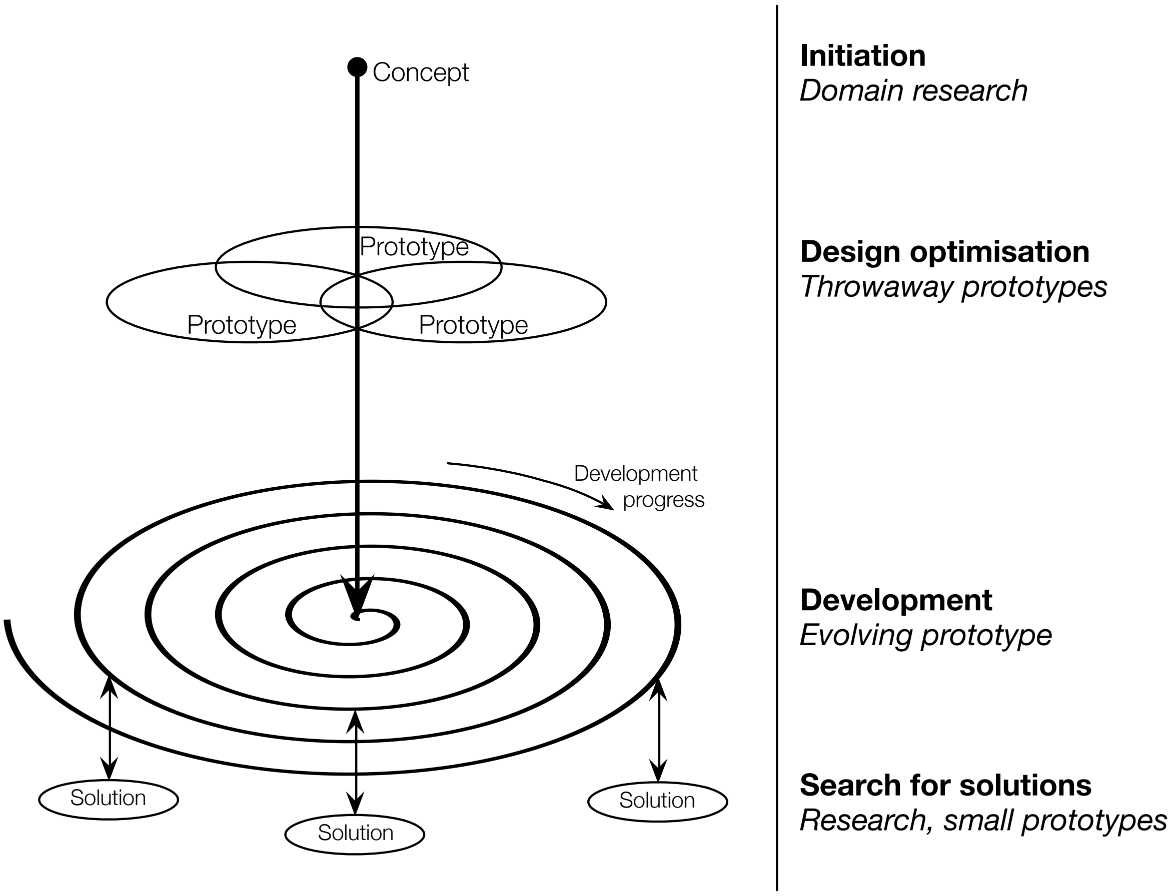


Figure 5.1: Project development process model

In addition to this the project could require research activities or exploratory development to be done before implementing a new set of features. In every case the research should be carefully planned including frequent communicating of the progress to simplify the associated decision making.

5.3 Development planning

Software development planning process, proposed by the methodology, is based on the following observations:

- It is hard to predict when the extra effort will be needed during the development of a project with high level of uncertainty. The usefulness of the buffering mechanism, used to manage the impact of variation and uncertainty in Critical Chain Project Management (CCPM) [115, 116] is limited to the completeness and correctness of identified critical chains.

- The knowledge about the project and project environment itself change throughout the development, leading to the shifted priorities of project activities in different project phases.
- The uncertainty level in the project is most likely to reduce to the end of the project.

The proposed software development methodology addresses these issues by introducing a planning approach, based on effort reservations for overcoming uncertainty. Once the initial project plan has been worked out and project effort estimated with the aid of a typical approach, an effort reserve threshold m is set up. The reserve threshold may vary depending on the accuracy and completeness of the initial requirements and the level of project risk and uncertainty. The threshold equal to 30% is taken for the instruction convenience: $m = 30\%$.

The reserve is meant to be gradually used during the project development for uncertainty reduction. To prevent spending all the reserve at once while being stuck on a single problem and thus ensure the delivery of most important and critical features of the software system on time, the reserve usage is being continuously monitored and adjusted if needed by means of activity prioritisation. Figure 5.2 shows task composition for each iteration depending on the intensity of using reserved effort.

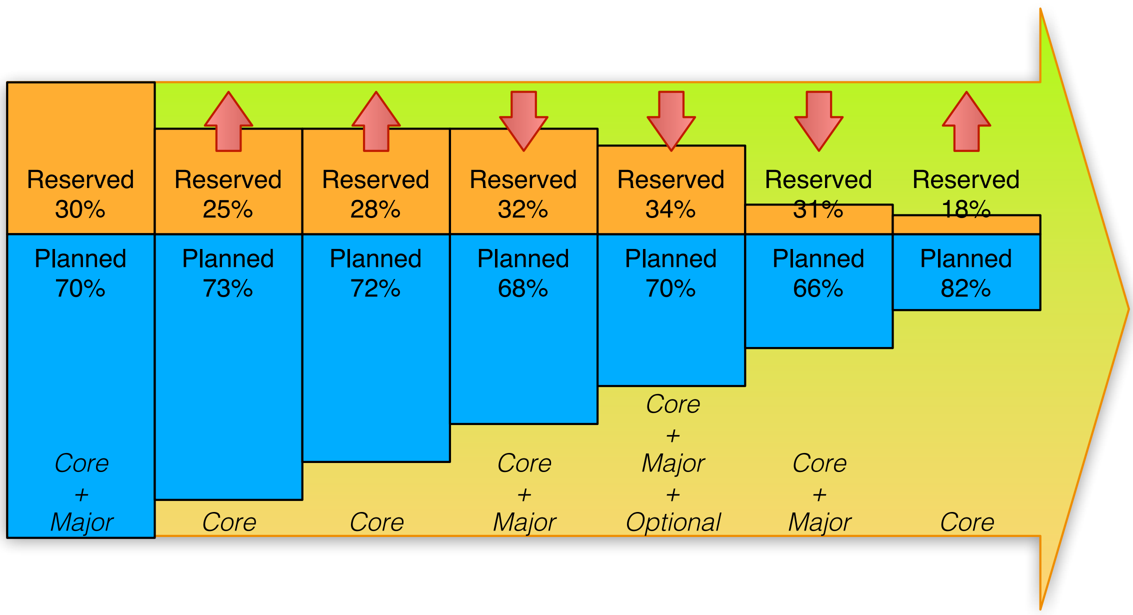


Figure 5.2: Planning task composition for each iteration depending on the intensity of using reserved effort

Normally the priority of project activities would be the following:

1. Reduce uncertainty
2. Implement complex tasks
3. Deliver business value
4. Refining and improving

But considering the reduction of uncertainty to the end of the project combined with increasing importance of business value delivery over the course of project development, as well as desire to avoid leaving complex task implementation to the end, the given action priority list needs to be altered for different project phases, as shown in Table 5.2.

Table 5.2: Task priority evolution throughout the project development

Project phase	**Core tasks**	**Major tasks**	**Optional tasks**
Initial phase	Reduce uncertainty		
Architecture	Reduce uncertainty	Implement complex tasks	Improvements and refinements
Incremental development. Beginning	Reduce uncertainty. Implement complex tasks	Deliver business value	Improvements and refinements
Incremental development. Middle	Implement complex tasks. Deliver business value	Reduce uncertainty	Improvements and refinements
Incremental development. End	Deliver business value	Improvements and refinements	Implement complex tasks

Therefore, the present methodology divides all project tasks into three categories listed in the order of execution priority:

1. *Core tasks* include the tasks, that must be done no matter what. These are planned activities for each phase of the project, such as specifying or refining requirements, implementation, refactoring, testing and debugging.

2. *Major tasks* include tasks, that would be very good to do. Tasks associated with project uncertainty, research activities and exploratory development are relevant to this category.

3. *Optional tasks* include tasks, that can be done if there is extra time in the project, for example, improving implemented features and algorithms.

Tasks from one, two or all three given categories can be scheduled for implementation within a single project iteration. The composition of tasks for every subsequent iteration is defined based on the currently available effort reserve c. A block scheme showing the process of decision making about the task composition for each iteration is shown in Figure 5.3.

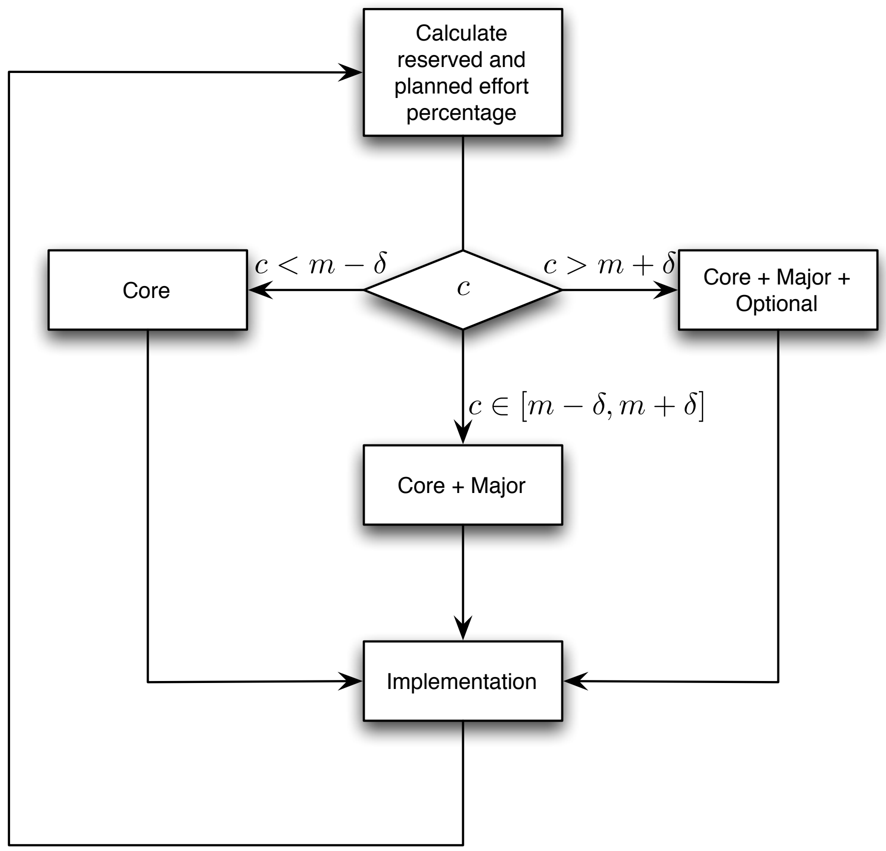


Figure 5.3: A block scheme showing the process of decision making about the task composition for each iteration. c - current reserve; m - reserve threshold; δ - lag size.

If the amplitude of current reserve c is within the defined lag δ around the project reserve threshold m, tasks from first two categories are scheduled for the iteration. The lag δ is introduced with the purpose to avoid unnecessary premature switching between accelerating and slowing down the reserve utilisation. The value of the lag δ would be normally about 10% of the reserve threshold value (it is 3% of total project effort for the given threshold m equal to 30%). In the case when the current reserve c is greater than the threshold m added to the lag δ, then there is extra time in the project for the implementation of optional tasks. If the current reserve c is less than the threshold m minus the lag δ, then the reserve is being used overly intensively and only core tasks should be scheduled for the current iteration. The need to increase or reduce the intensity of reserve utilisation depending on the given reserve size is depicted in Figure 5.2 with red arrows.

5.4 Software design optimisation

5.4.1 Web-based CAD/CAM software product concept

In order to develop a product effectively it is essential to have a clear vision of what this product represents. Figure 5.4 contains a mind-map of different considerations related to a Web-based CAD/CAM software product. It contains aspects common to many software products, as well as those specific to Web-based applications and CAD/CAM systems. The diagram could serve as a roadmap for creating initial architectural design, by the way of highlighting common decision points of Web-based CAD/CAM development that need to be kept in mind.

Any software product is characterised by application size and complexity, the scope of functionality to be implemented and business processes it encompasses. The functionality and behaviour of the software product is defined in its source code; and the overall application maintainability is greatly impacted by the source code design, id est, the use of patterns, naming and coding conventions and explanatory commentary. The development of a software product essentially is writing its source code, therefore the reuse of legacy code from external sources can significantly reduce the effort required. The amount of third-party code in contemporary software projects can be more than three times as much as the in-house developed code, and also that in-house code can contain about one third part of legacy code transferred from previous projects [117].

The quality and life cycle of a future-proof software solution is greatly conditioned by technologies implemented by the working team, following standards and recommendations, as well as being aware of new technological developments and trends. The architecture of the software product defines its capabilities for evolution and maintenance throughout the whole life cycle, therefore a clear understanding of the application usage context is required. This includes not only user experience, backgrounds and skills, but also the environment and conditions, in which the software product is supposed to be exploited: software and hardware technologies, any geographical and business conditions, as well as perspectives for further development and integration.

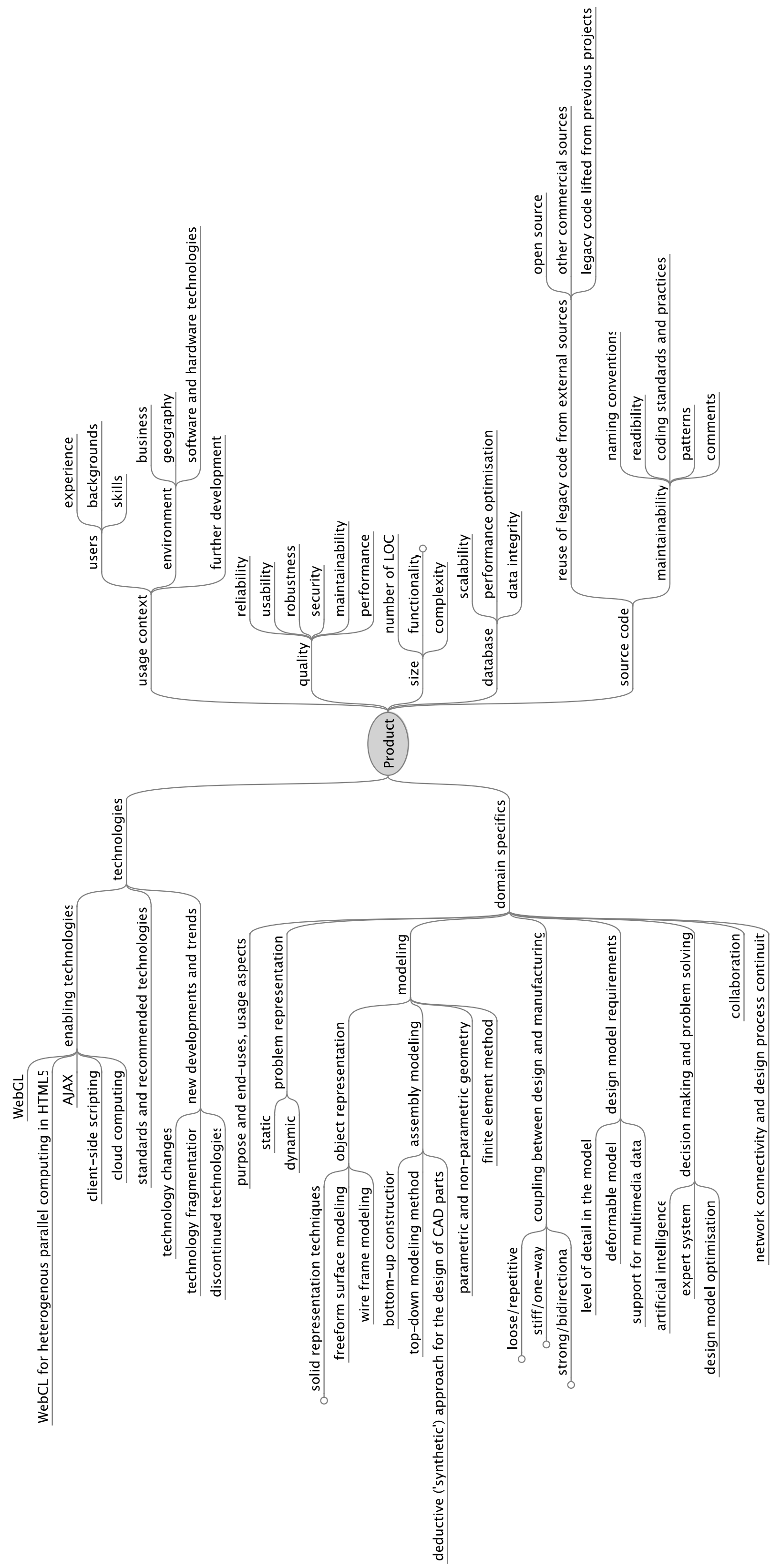


Figure 5.4: Web-based CAD/CAM software product considerations

As it was mentioned in the beginning of this section, Web-based CAD/CAM software products have also some domain specific features to be aware of when developing this kind of applications. First of all this includes software product usage aspects, it is the purpose of the application and its end-uses. Secondly, the implementation of a CAD/CAM system depends on the representation of the problem it tries to solve – either static or dynamic.

In order to aid CAD/CAM process a dedicated software application should be able to support required workflow. This assumes implementation of a relevant modelling approach, using parametric or non-parametric geometry and appropriate object representation techniques and assembly modelling methods. Design models can have different level of detail for different end-uses, and the model itself can have the ability to undergo deformations. Finite element method can be employed to extend the capabilities of a CAD/CAM application in terms of control over analytical and modelling complexity and provide options for managing time and accuracy required for engineering computations [118].

For software systems supporting complete CAD/CAM workflow the coupling between design and manufacturing can be implemented in three different modes [119]:

- loose/repetitive, providing maximum design flexibility, but redesigns are slow and expensive,
- stiff/one-way, providing less design freedom, but guaranteed manufacturing,
- strong/bidirectional, providing moderately flexible design and guaranteed manufacturing, sacrificing some design freedom.

CAD/CAM software product can implement design optimisation techniques – single- or multi-objective, as well as employ Artificial Intelligence (AI) elements and expert system technology to assist the user in decision making by means of using knowledge base and inference engines. The primary concerns of AI application for CAD/CAM purposes is exploring the formal representation of design knowledge and development of techniques for reasoning using this knowledge [120]. AI technologies may be relevant for solving design and manufacturing problems, which are classified in the literature [120] as follows:

- decomposition of complex tasks, which means dividing the design problem or the design itself into smaller elements. Depending on the direction, in which the design process progresses, the solution can employ top-down or bottom-up approach, or combine both;

- plan refinement, based on reducing the design problem and deciding on which design refinements to perform next as the design progresses and the design model is better specified;

- constraint-based reasoning, guiding the design specification process through a network of constraints associated with design attributes;

- case-based reasoning, which includes decision making techniques based on the previously accumulated experience for solving similar problems;

- consistency maintenance or truth maintenance, aimed to solve the problem of maintaining context related data and values assigned to design attributes while progressing throughout the design process.

5.4.2 Critical design decision points

In a Web-based CAD/CAM system the most critical design decision points, affecting overall system performance and usefulness, comprise:

1. application interactivity,

2. task distribution between the server and the client,

3. multiuser interaction.

Interactive capabilities of the system define its ability to accept and respond to input from the users. Solving interactivity issues in Web-based CAD/CAM software is associated with defining the client side functionality and mechanisms for graphics manipulations, handling application events, optimising system responsiveness and managing data input/output effectively (see Figure 5.5).

Addressing network connectivity and design process continuity challenges assumes managing availability, capacity and client-server interaction intensity, estimating application risks and building in recovery mechanisms, and also involving data and security management.

The problem of task distribution between the client and the server greatly affects Web-based CAD/CAM system performance. Figure 5.6 shows a possible way of distributing various CAD/CAM operations between the server and the client. The operations like manipulations with graphics, data input, input data validation and displaying design results could be performed on the client, while computation-intensive tasks, such as

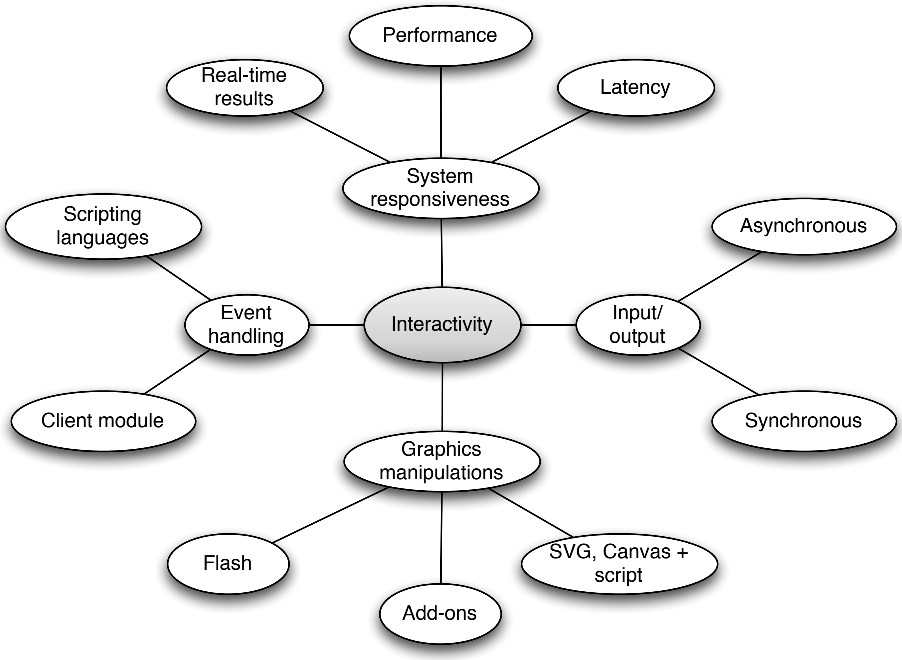


Figure 5.5: Web-based CAD/CAM system interactivity

design data calculations, optimisation, data mining or output file generation may be reasonable to implement on the server side.

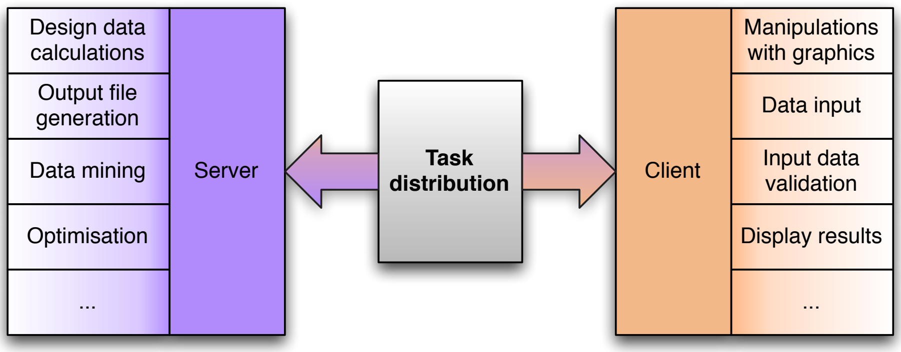


Figure 5.6: Task distribution between the server and the client

One of the main purposes for the development of CAD/CAM software based on Web technologies is the inherent capabilities of Web applications for multiuser interaction, enabling wide opportunities for collaboration and document management. Implementation of collaboration features in a Web-based CAD/CAM system presumes establishing mechanisms for resource and knowledge sharing. Multiple concurrent input and output is a concern of design for capabilities supporting simultaneous work on a particular design.

Interaction circumstances may vary for different applications and depend on requirements for coincidence or difference in time and place (see Figure 5.7).

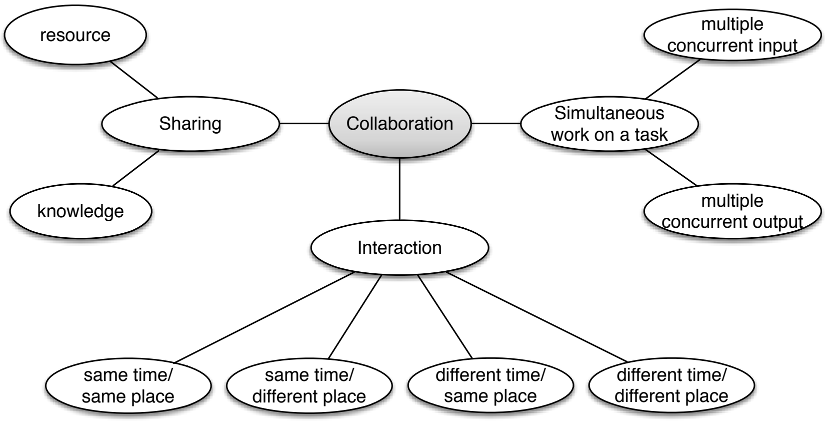


Figure 5.7: Software collaboration concerns (based on [121])

A CAD/CAM system, implemented using Web-based technologies, enables wide opportunities for the integrated document management in the company, encompassing business workflows and providing capabilities for document drafting, editing, versioning, reviewing, distributing and storage. In this context CAD/CAM application design incorporates defining interfaces to other systems.

Based on the initial domain research and defined key scenarios and requirements, the application infrastructure and deployment constraints could be sketched. Developing one or several throwaway prototypes addressing critical design decision points discussed above could help to investigate the most appropriate technologies and application infrastructure, covering some of the most critical risks associated with the design of Web-based CAD/CAM system.

5.5 Addressing Web-based CAD/CAM development concerns

When developing a software product for industrial usage purposes the quality of deliverables is expected to meet certain requirements for application reliability, usability, robustness, performance, maintainability and security. The software developers have to be also conscious of database capabilities, performance requirements, scalability and data integrity.

5.5.1 Performance

Software performance may be essential for solving business tasks as it impacts user experience and productivity. As reflected in Figure 5.8, to address performance concerns software engineering best practices include actions for discovery of production environment, environment virtualisation and testing, and also analysis and optimisation of performance issues [122].

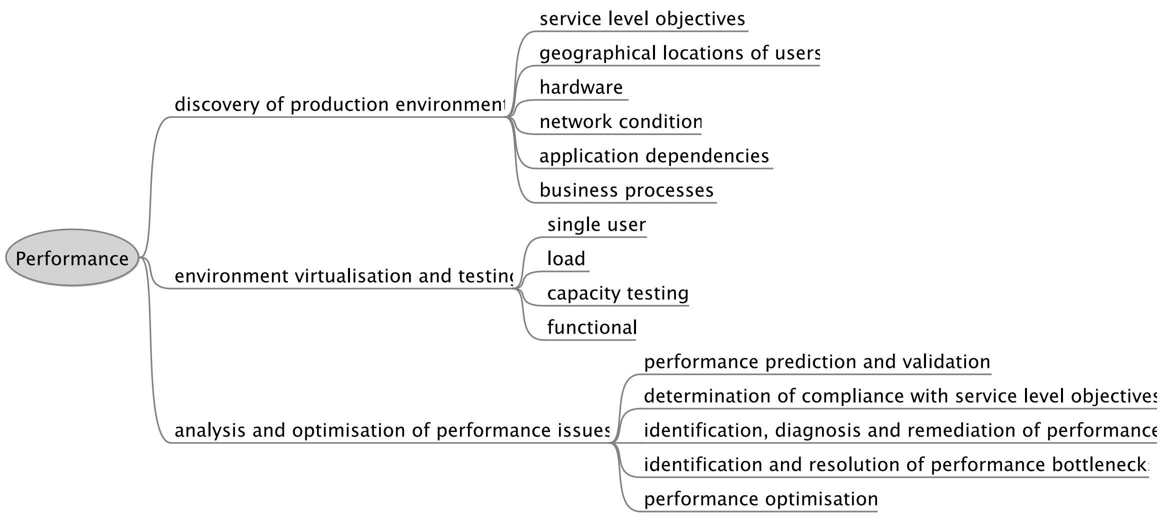


Figure 5.8: Addressing software performance concerns (based on [122, 123])

In accordance with best practices for software performance engineering [123] it is advised to identify and focus on critical parts of the software and ensure that the architecture is able to support performance objectives. Software architecture and design alternatives can be evaluated using quantitative performance models before committing to the actual development phase. For better management of uncertainty, the performance models can develop from simple to more detailed as the knowledge of the software increases throughout the project development.

5.5.2 Robustness

Developing robust software involves activities aimed at ensuring its resilience to stress conditions and ability to cope with errors during system operation. Addressing Web-based CAD/CAM software robustness is associated with a number of concerns shown in Figure 5.9.

Design for software fault-tolerance is based on anticipating exceptional conditions and can incorporate such measures, as error and exception handling, self-stabilisation mechanisms, duplication, replication, redundancy and diversity. Following robust software

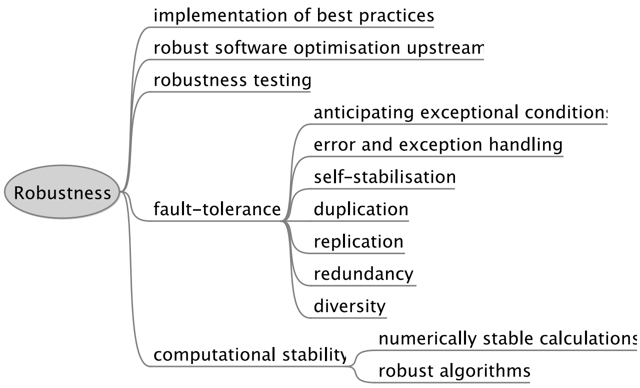


Figure 5.9: Addressing software robustness concerns (based on [124, 125, 111])

development best practices, upstreaming software design optimisation and robustness testing is essential for successful development of robust software products. Computational stability plays an important role in ensuring robustness of scientifically intensive and engineering applications, and in particular CAD/CAM software.

5.5.3 Compatibility and interoperability

Compatibility and interoperability define software ability to operate with other products and are important for the development of production-ready Web-based CAD/CAM applications. In Figure 5.10 software compatibility concerns are summarised based on previous software engineering experience [126, 127, 128]. Those incorporate software environment, mode of operation, compatibility of software versions, components and interfaces, performance related issues.

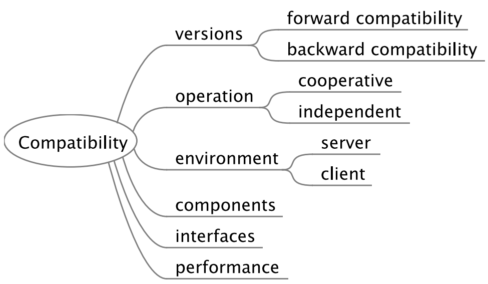


Figure 5.10: Key considerations for software compatibility (based on [126, 127, 128])

When designing Web-based CAD/CAM software for compatibility, both client and server side compatibility requirements need to be taken into account. Figure 5.11 shows client side compatibility aspects in detail. The aspects can be divided in two parts: relevant to the platform choice and relevant to the software implementation. The software implementation basically is a choice between browser-based and client-based technologies, each of them having specific compatibility concerns.

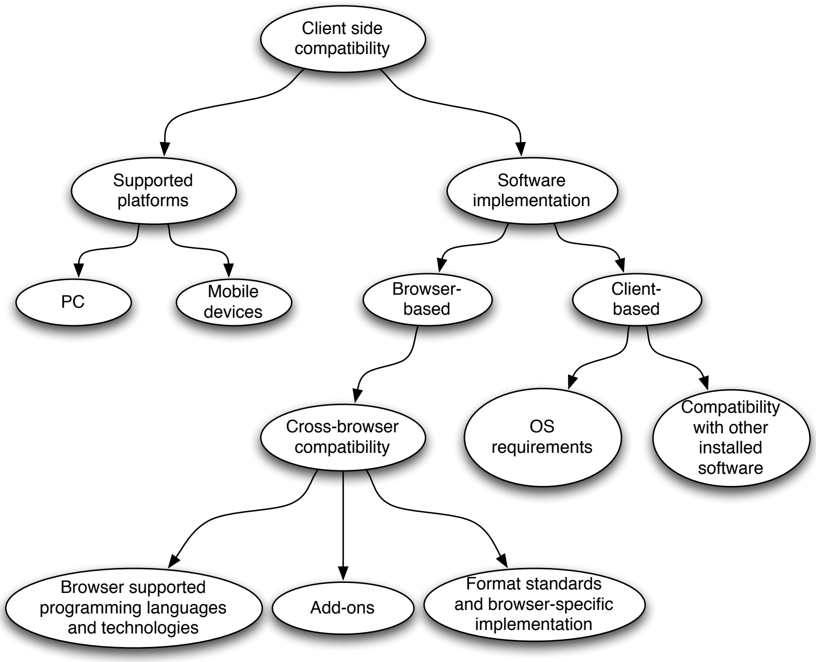


Figure 5.11: Client side compatibility

As CAD/CAM applications can be used in conjunction with PLM and other company's software to establish better support for business processes, it is important to ensure good interoperability of Web-based CAD/CAM system. Designing interoperable software, which operates successfully by exchanging information internally and externally incorporates such aspects, as standardisation and formalisation of communications, interfaces and data formats [128].

5.5.4 Flexibility

Flexibility is an important characteristic of quality software product as it allows to reduce amount of work associated with modifying the software to establish changed or new functionality. As shown in Figure 5.12, internal, external and runtime flexibility could be distinguished [129]. Internally flexible software allows changing or extending functionality without application redesign by making simple adaptations and modifications to the source code. Implementation simplicity, modular design, use of coding conventions and patterns, loose coupling of software components, OOP and structured programming practices all contribute to application internal flexibility and lead to better quality and maintainability. In turn, software products, that incorporate reusable components and can be used as a library or framework, possess external flexibility.

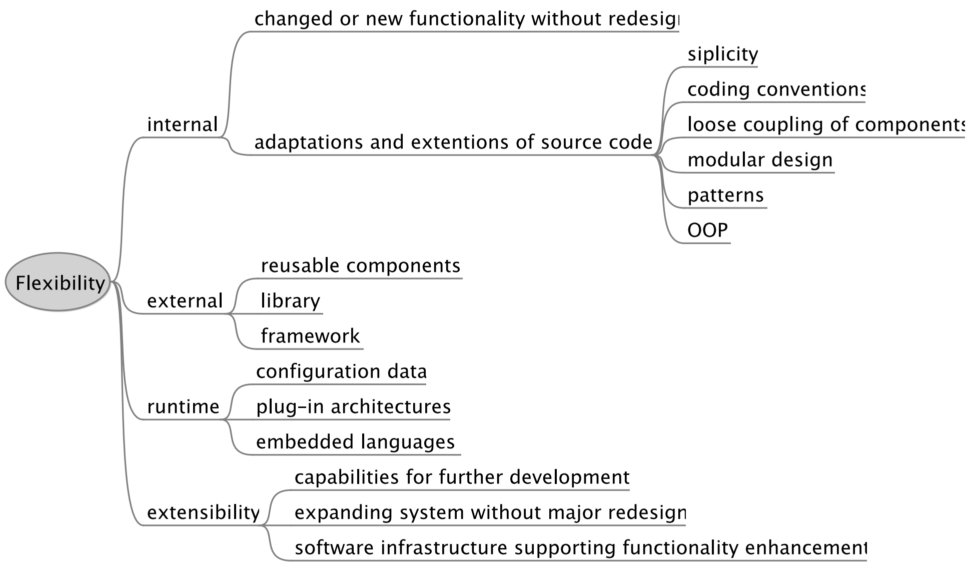


Figure 5.12: Addressing software flexibility concerns (based on [129, 130])

Runtime flexibility defines the ability of the software to establish changed functionality without the need to make changes to its source code. Software runtime flexibility could be achieved with the use of configuration data, plugin architectures and embedded languages [129].

For creating easily maintainable and future-proof software products extensible architectures are implemented, providing the design principles to support possible future system extensions without making major changes to its infrastructure. Software products with extensible architectures usually have built in some excess capabilities, that are not used in current delivery, but may be needed in the future. Although there is a chance that these excess infrastructure capabilities may never be used, creating extensible architectures may be beneficial as it allows avoiding implementation of a number of different possible solutions for any anticipated requirements [130].

5.5.5 Scalability

Scalability is an important design concern of Web-based CAD/CAM systems, as it defines the ability of distributed application to handle increases in load or be readily enlarged without adverse impact on the performance of the system. The system scalability is not something, that can be added later during the development or deployment, therefore it must be allowed for in the design stage.

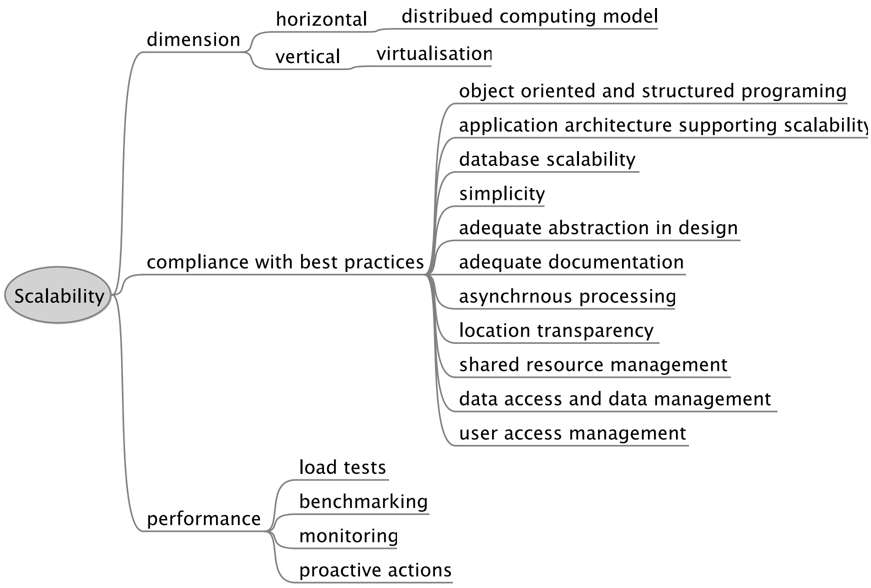


Figure 5.13: Distributed systems scalability aspects (based on [128, 131, 132])

Figure 5.13 encompasses various scalability aspects associated with the development of Web-based CAD/CAM systems. First of all it is worth mentioning that scalability of a system could be achieved in two ways: scaling vertically and scaling horizontally. The first method assumes upgrading hardware to allow for an increase in capacity without changing the application source code. In turn, scaling horizontally allows distributing the processing load across more than one server by dedicating several machines to a common task [128].

The problem of creating scalable software can be addressed by following good software design and development practices, including applying object oriented and structured programming practices, processing parallelisation and asynchronous processing, creating scalable database and architecture supporting scalability, ensuring simplicity and adequate level of abstraction in design. Location transparency may be critical to successfully scale application horizontally. Software capabilities for scalability also depend on management of shared resources, user access management, data access and data management.

Performance and scalability are especially important for applications incorporating algorithmic, transactional or large scale data processing requiring high execution speed. Performing load tests, monitoring, benchmarking and profiling the application are critical for the discovery of potential problems and performance bottlenecks requiring immediate resolution or proactive actions before the system becomes unusable [131].

Specifying target average and peak performance and load and stipulating acceptable limits for application scalability, response time, latency etc., as early in the project as possible is critical for selecting a successful approach for addressing application scalability concerns [132].

5.5.6 Reliability

Addressing software reliability concerns is about ensuring the ability of the software to perform a required function under specified conditions. As shown in Figure 5.14, on one hand software reliability is associated with avoiding unexpected behaviour by error-free software development, using error-prone programming constructs, applying fault tolerance techniques,subtypes and runtime assertions, error and exception handling, resource bounds management, safe multi-threading and performing software tests [133]. In a Web-based CAD/CAM system sources of failure include host, network, browser, user error or source content failure [134].

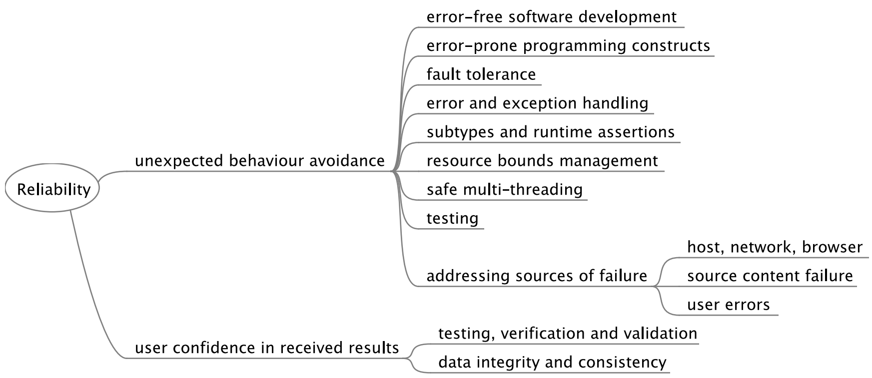


Figure 5.14: Addressing software reliability concerns (based on [134, 133])

On the other hand reliability is defined by user confidence in results, received when working with the application. This could be achieved by ensuring data integrity and consistency, extensive testing, verification and validation of software produced results.

5.5.7 Security

To ensure the ability of an industrial Web-based CAD/CAM system to protect its resources a range of measures could be undertaken. Those are summarised in Figure 5.15 and include access management, data protection, following software security best practices and educating users.

Managing access to resources, programs and system functions is based on the concepts of authentication and authorisation, controlling permissions and roles. Data protection measures are about ensuring data integrity and confidentiality, whether the data is in

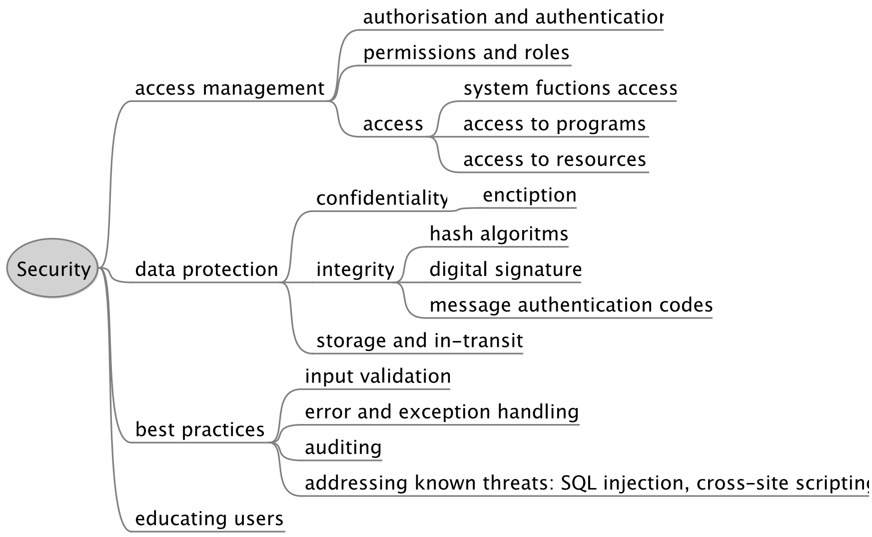


Figure 5.15: Addressing software security concerns (based on [135])

transit or stored. Developing a secure software application could be facilitated by applying security best practices, such as input validation, error and exception handling, auditing and measures for addressing known threats, like Structured Query Language (SQL) injection or cross-site scripting.

5.5.8 Usability

Usability and user productivity are among the most critical features of industrial applications as they directly affect efficiency and effectivity of the associated business processes. The key considerations for software usability are represented in Figure 5.16. Although software product functionality and the workflow it supports have impact on the application usefulness and the amount of effort the users expend when using it to solve their tasks, to the greatest extent software usability is defined by the user interface. The user interface usability encompasses various aspects of user interaction with the system, such as how simple and consistent is the user interface and how easy it is to discover, learn and remember how to use it. To meet usability requirements the user interface has to be efficient, pleasant to work with, it should also provide full user control over the system, with informative and sensible error messages and options for error recovery.

As the software is created to be usable for a certain group of users, their knowledge, backgrounds, skills and abilities, have to be taken into account when developing the user interface. In this context user feedback can not be underestimated for the user interface acceptance by the end users. The user feedback could be obtained in various

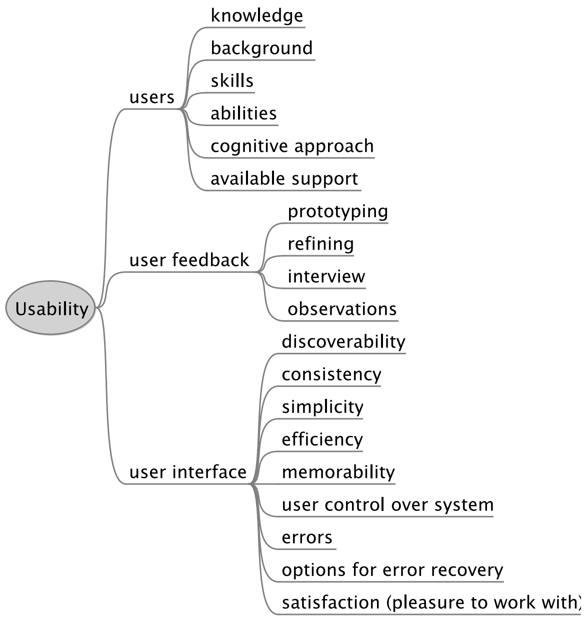


Figure 5.16: Key considerations for software usability (base on [136])

ways with the use of prototyping, conducting workshops with potential users and making observations of how they interact with the system, and also interviewing users about their experiences with the user interface.

5.5.9 Maintainability

The ease of fixing software errors, making functional modifications and meeting new business requirements is defined by application maintainability. Accomplishing good software maintainability is associated with its design, the condition of the source code and documentation (see Figure 5.17) and is affected by the overall quality of the system.

Software design facilitating good maintainability should consider multi-layer structure and independence of hardware, operating system, middleware, software and database components. The design should also follow a strategy for avoiding complexity in interfaces, algorithms, transactions and application design, considering coupling between components and applying good programming practices. Maintainability is affected by application modularity, component reusability and understandability and allows to reduce effort required for making changes and testing, porting the system or transferring from one development team to another [137].

The quality of the source code is an important concern for the software product maintainability. This includes code readability, its structure and organisation of source code files, the amount and level over dynamically generated code, as well as the amount of

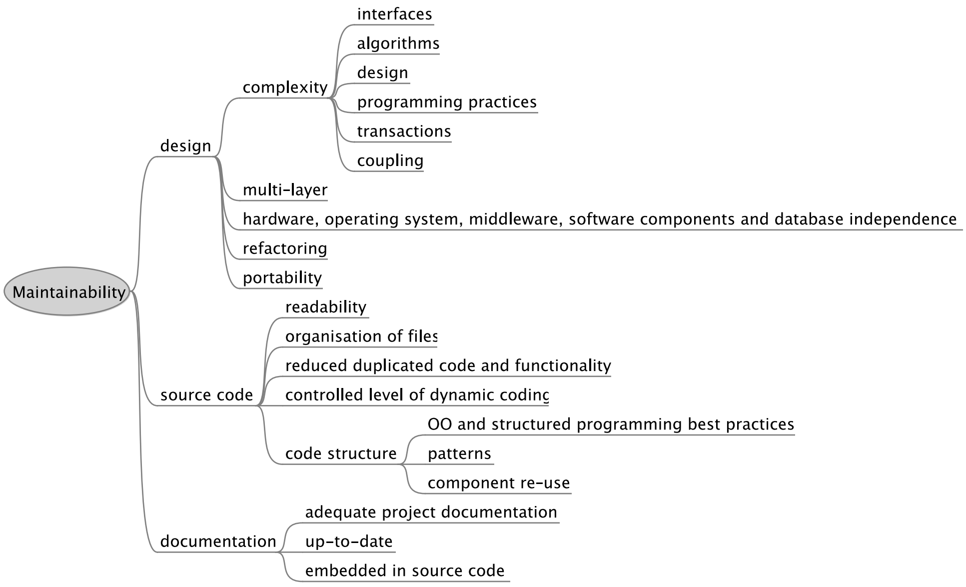


Figure 5.17: Addressing software maintainability concerns (based on [137])

duplicated code and functionality. Availability of adequate and up-to-date documentation also greatly impacts the maintainability and embedding it in the application source code is desired.

5.6 Requirements elicitation

As it was mentioned in section 4.2, Web-based CAD/CAM software development brings together people with very different expertise and backgrounds, magnifying understanding problems associated with the requirements elicitation. Confusing information shared between stakeholders and the lack of understanding aggravates project scope and requirements volatility problems, conditioned by the business application complexity and the specifics of the expertise and knowledge it involves.

In the beginning of the project it is essential to assess the business and technical feasibility for the proposed system, and also gain as clear understanding about what the final software product should represent and as precise comprehension of the project scope as possible. For this purpose the software concept is described, key use cases are identified and initial requirements are obtained. The quality of these initial requirements, as well as of those, that are defined later during the development, among the rest, greatly depends on the developers competence in domain specific questions.

The developers competence in the industry specific areas could be achieved by organising sessions for transferring relevant industrial knowledge to the development teams.

Learning domain specific topics is important for building understanding about underlying principles and technologies and ensuring that developers and stakeholders speak the same language. Writing and communicating definitions of terms also makes sense.

The methods for requirements elicitation can be used different, from semi-structured interviews with stakeholders and discovering usage scenarios to making observations of how users work, developing software prototypes and obtaining feedback by demonstrations and conducting workshops. Stakeholders could be also involved in modelling business requirements workflows and user interactions using sketches and simple free-form diagrams.

For the software requirements elicitation it is important to ensure the involvement of the stakeholders, that are competent in relevant questions. Also multiple people participation in requirements elicitation is preferred to consider different opinions and points of view. The rationale, implementation priority and the acceptance criteria should be provided for each requirement. With the help of exploratory development and prototyping initial set of requirements is extended and ambiguous requirements are refined.

5.7 Overcoming uncertainty

Software development projects involving novelty and creativity are usually conjugated with high risks. Provided with some previous experience, the risk of failure can be possibly measured for a similar project. Technical novelty makes the task of project risk assessment more challenging, due to the inability to accurately describe technical risk elements and factors, which can influence this risk.

A number of methodologies were developed to aid technical risk assessment, commonly based on either probabilistic analysis, or anchored scales approach. Probabilistic Risk Assessment (PRA) [138] is based on a systematic and comprehensive risk evaluation, describing the risk with the product of the probability of occurrence and severity of the possible consequences. PRA involves deterministic evaluations, event sequence diagrams analysis and quantification using probabilistic or statistical methods.

Anchored scales approach [139] analyses project risks and rewards using the probability factors and reward factors that are appropriate to a particular business area. To aid the evaluation consistency the approach proposes to "anchor" the measures with unambiguous descriptions of the relevant risk levels. The scores are further aggregated mathematically with an option for weighting importance as well.

Although many recognise that neither method is very successful [140], identifying the sources of risk and systematical approach to risk management facilitate the project success.

When dealing with the lack of information for predicting the outcome of a software development project, it is essential to recognise, what are the causes for the uncertainty. The selection of risk management strategy depending on the type of uncertainty prevailing in the project has been discussed in the literature [141, 142, 143] and different classifications of uncertainty factors have been established. For convenience, in regards to the software development context, the proposed methodology considers three common factors, that make the task of project planning more challenging:

- *Uncertainty caused by dynamics.* Dynamics here is related to any changes, that can impact project development, such as changing requirements or rapidly changing technologies.

- *modelling uncertainty.* Missing knowledge about the behaviour of implemented models within the system being developed. For example, uncertainty related to the performance of an analytical model when implementing it for solving certain tasks using selected technologies.

- *Variations.* Small deviations in the input parameters that can not be foreseen. The result (whether it would be flexibility of the design or performance of the final product) should remain acceptable despite the variations in input parameters. For example, the performance of an analytical model implemented in the system would be still acceptable, if the number of calculations would be necessary to increase by 10%.

The ability to make accurate and realistic software development plans can be improved by undertaking various activities, aimed at the reduction of uncertainty in the project. The actions can follow one of two possible strategies:

- Improve adaptation to the change to minimise effort needed to adjust to the current situation,

- Minimise the impact of the change on the project.

For the implementation of the first strategy frequent reviews and feedback is vital. To achieve this the proposed methodology utilises iterative development model. The second strategy is based mainly on decisions related to the design of the software system. For example, when dealing with rapidly changing technologies following standards can lower the level of associated uncertainty, as standards do not change often. In the case of changes in user requirements, such measures as using patterns, OOP, application modularisation and different software abstraction levels can be very helpful as those can improve response to the change by providing mechanisms reducing associated effort.

The problem of modelling uncertainty can be addressed by applying throwaway prototyping. This approach suggests making simple prototypes in short time to check how well the particular design, model or algorithm is able to solve the problem.

The key to the management of variations in the project is search for robust solutions, which could be achieved by investing in the development of a proper design, producing good structured code, implementing different abstraction levels and choosing technologies allowing for some reserve in performance.

5.8 Resource allocation and collaboration

As mentioned in section 2.4.2, it is very likely that a Web-based CAD/CAM software development project will require specifically created development team, involving people with expertise in different fields. Also temporary participation of external experts may be required for solving some tasks and problems encountered during the development. These circumstances create additional challenges for resource allocation and project planning.

The main challenges are associated with availability of involved experts, commitment of stakeholders and team collaboration capabilities and productivity. Project schedule flexibility governed by time reserve management and prioritised task execution are intended to facilitate resolving expert availability issues.

Despite having a common goal challenges associated with stakeholder commitment may arise in a software development project due to possible conflicts in stakeholder interests. Methods for dealing with this successfully and ensuring and maintaining stakeholder commitment throughout the project duration incorporate identifying and prioritising stakeholders based on their commitment and influence on the project, understanding each stakeholder motivation and agreeing on their role and involvement [144]. For the success of the project portfolio management it is also important to choose optimal intensity of engagement for each internal stakeholder specifically for each project phase, as the effect from intensive engagement of stakeholders performing different roles is not always clearly positive and varies from phase to phase [145].

Knowledge exchange and team collaboration are critical attributes of science-intensive and specialist CAD/CAM software development process, impacting productivity and personal development of team members, facilitating learning, decision making and innovation [146]. To enable team knowledge base construction and collaborative problem solving it is required to ensure collaboration capabilities, comprising communication infrastructure, conventions and support for information exchange, infrastructure for team collaboration and negotiation, as well as infrastructure for outcome evaluation and revision. The employed collaboration technology influences team collaboration behaviour and performance, therefore implementation of advanced collaboration concepts and technologies is preferred for enhanced productivity [147].

5.9 Applicability

The purpose of this methodology is to provide guidance on the development of complex science-intensive software to industrial quality standards in a time and cost effective manner. The methodology primarily aimed to support the development of Web-based CAD/CAM systems, although it has many aspects common for other software too.

The main prerequisites for applying the proposed methodology are the following:

- The project combines CSE and commercial software features.
  - Correct and efficient operation of the software is critical.
  - Usability, maintainability and extensibility greatly impact the success of the project.
- The software development process is rather exploratory, than predictable
  - The development presumes the necessity to conduct a number of small studies, creating and verifying prototypes of separate software modules during the development.
  - The strict adherence to formal practices may disable creativity and innovation or lead to excessive overheads.
- It is hard or impossible to identify the complete set of user requirements in the beginning of the project.

  - Project scale is too big to identify all requirements at once. Attempting to do so would cause 'paralysis of analysis' or lead to soon degradation of software design.
  - Some requirements can be specified only after a certain amount of functionality has been implemented.
  - Requirements or user vision of final product significantly change throughout the development.
  - The degree of specification ambiguity is high, for example, due to the diversity of expertise of the parties, involved in the process.

- Because of the previous points, it is impossible to create a predictable schedule and budget.

The methodology builds on features of widely used software models and proposes an optimised model for agile development of Web-based CAD/CAM systems and industrial science-intensive applications.

The methodology incorporates several unique features, such as:

- Upstreaming design optimisation.
- Minimising the necessity to deal with the moment of inertia in incremental development model.
- Supporting research activities and exploratory development, while staying focused on requirements.
- An approach to planning unpredictable software development projects.

The methodology is largely unsuitable for the projects that possess the following features:

- Small project scope with easily obtainable and well defined requirements.
- Predictable development process that could be accurately scheduled using standard time and effort estimation techniques.
- Inability to decompose the project in smaller parts, which is necessary for the successful application of iterative development approach.

5.10 Projected benefits from applying the methodology

Applying the proposed methodology for Web-based CAD/CAM system development could benefit the project by:

- Providing an approach for planning and dealing with uncertainty and thus achieving project goals and attaining user/customer satisfaction with the software product.
- Describing an approach for making key design decisions and selecting optimal application architecture early in the project to minimise total development efforts, create future-proof solution and ensure system maintainability.
- Establishing a roadmap for design and delivery of quality software.
- Providing project development model supporting early delivery and return of investments, as well as facilitating user involvement in the project.
- Describing requirements elicitation process facilitating accumulated industry specific knowledge and experience capture, as well as discovery of new solutions and generating new knowledge.
- Providing guidelines for creating adequate and useful documentation reflecting the way the software actually works.

5.11 Summary

Based on the prerequisites for the Web-based CAD/CAM software development methodology, formulated at the end of Chapter 4, and analysis of software engineering experience, provided in Section 2.4, a set of guidelines for creating this kind of applications are proposed. The description of these guidelines consists of establishing major considerations and key principles to aid the software development process and address the associated challenges, structuring the development process and setting objectives for different phases, explaining the approach to planning, design optimisation, requirements elicitation, strategies for overcoming uncertainty, resource allocation and collaboration, as well as detailing how the Web-based CAD/CAM software development concerns could be addressed to ensure application reliability, usability, robustness, performance, maintainability and security, required for the system use in production.

The described methodology for creating Web-based CAD/CAM systems and industrial science-intensive applications builds on features of widely used software models and proposes an optimised model for agile development, also incorporating several unique features, such as upstreaming design optimisation, minimising the necessity to deal with the moment of inertia in incremental development model, supporting research activities and exploratory development, while staying focused on requirements. An approach to planning unpredictable software development process is also proposed.

The methodology is intended for the development of complex science-intensive software to industrial quality standards in a time and cost effective manner. Although being primarily aimed to support the development of Web-based CAD/CAM systems, it has many aspects common for other software too. Presumably, the methodology could be successfully applied for the development of software, that has features of CSE and commercial applications, has incomplete user requirements specification, assumes less predictable and exploratory software development process and difficulty of producing a predictable schedule and budget.

Chapter 6

CAD/CAM system for gear shaper cutters

In this part of the research, the proposed methodology is applied for the development of two case studies. The choice of the case studies is based on the industrial focus of this research. The first case study includes the development of a Web-based CAD/CAM system for involute spur gear shaper cutters. The second case study describes the development of a Web-based CNC code editor for online modification of the profile for manufacturing gear shaper cutters. Although the case studies are linked to gear cutting tool manufacturing, the methodology can be used for the development of CAD/CAM software in different context.

The Chapter structure is organised based on the description of the methodology for Web-based CAD/CAM software, provided in Chapter 5, covering all key aspects of case study development from software concept and requirements elicitation to planning the development, resource allocation and creating optimal design of the software. A short description of the software architecture, implemented technologies and user interface is also provided. Then, challenges faced during the case study development and implemented solutions are overviewed. Finally, impact of the development of Web-based CAD/CAM for GSCs is discussed and a summary is provided at the end of the Chapter.

6.1 Design and manufacture of gear shaper cutters

Gear shaper cutter industry has been selected for the development of case studies for this research due to the following reasons:

- Gear shaper cutter CAD/CAM system could serve as a typical example of a specialised industrial CAD/CAM application, requiring expertise in applied mechanics, mathematical modelling, algorithm development and design data visualisation;
- There are no solid software solution specialised for gear shaper cutter CAD/CAM;
- Web-technology could provide significant benefits for the industry in terms of world wide collaboration and improved software maintenance and manageability.

Gear shaper cutters mentioned in the case studies could be described as mechanical tools used to produce gears of various kinds. *Generating cutting* is the most widely used process for gear manufacturing due to its ability to mass-produce high performance internal and external spur and helical gears of different sizes. In order to cut the material of blank without rubbing, it correct mesh between cutter and gear is ensured by involute profile. A gear shaper cutter might have tooth profile modifications required to produce gear with certain special properties [148]. The basic modifications to the tooth profile include chamfer, fillet, protuberance, root and tip reliefs of different configurations. In addition some other parameters of the cutter could be adjusted to get a better tool life or better manufacturability. This leads to a highly parametrised design process involving complex gearing geometry.

The underlying principle of gear design is the fact that two toothed bodies moving relative to each other in constant mesh have their conjugate profiles in determined relationship [148]. The rules of that relationship are well described in the literature [149, 150, 151].

"The task of gear shaper cutter design could not be solved by off-the-shelf CAD package due to a highly parametrised design process that requires extensive computations and nature of the generating process. In turn, specialised gear shaper cutter design software, which could be used in production, is not available in the market. Software tools currently used for gear shaper cutter design are deprecated and require replacement able to work with modern hardware" [20]. Development of new Web-based CAD/CAM software for GSCs is vital for the business, and implementation of Web-based technology promises improvements of their competitiveness in many ways, from enhanced scalability and manageability to linking CAD/CAM with other business operations.

6.2 Case study description

CAD/CAM system for gear shaper cutters is developed to support cylindrical spur and helical gear shaper cutter design and manufacture. The aim of the CAD/CAM system development is to fully replace currently used deprecated HP-based software program for gear shaper cutter design and enable utilising modern data storage and retrieval technologies.

Analysis of the business environment shows, that a Web-based solution for the new software is more future-proof than stand-alone implementation. As cutter, gear and basic rack profiles are defined in two dimensions, it is possible to develop a platform independent application with the use of freely available components and standards.

New system is a solid software solution encompassing functionality of dedicated tools for design of different kinds of gear cutting tools. The system includes an option to print out design data and technical drawings at a user defined scale. Initially the system is not intended to be paperless, because of the need to print out internal job tickets, design drafts and technical drawings. But design master gear data should be stored electronically, and paper copy created only if needed. The new CAD/CAM system allows creating new gear cutter design, reviewing and modifying previously saved designs and deleting design data from the database. System security is realised with user access control through user accounts, protecting access to the system with user passwords.

For the case study of creating the Web-based CAD/CAM system for gear shaper cutters functional and user requirements are not defined completely and accurately at the beginning of the development, but rather are expected to be gradually added and refined. This results in hardly predictable, exploratory software development process. The case study is also limited in time and people resources and thus could possibly benefit from the incremental development. In other words, the case study fits well the main prerequisites for the applying the proposed methodology.

6.2.1 Software concept and purpose

The aim of the initiative is to develop a software system, that will deliver the best possible support for Dathan to provide a flexible and rapid design and manufacture service required by its customers. The new CAD/CAM system is expected to improve competitiveness of the company by means of increasing the effectivity of designers' work, accumulating and re-using knowledge about the designed products, as well as implementing improved design methods and techniques and increasing precision and quality of technical drawings.

The new CAD/CAM system is intended to support complete computerisation of the gear shaper cutter design process, thus minimising the number of errors in the data input operations between the different stages of the design process. The Web-based nature of the application will enable accessing and using it from any place with the Internet connection.

The specific objectives of developing the Web-based CAD/CAM system for GSCs include:

1. Replace the existing HP-based CAD/CAM software.
2. Improve current methods and techniques of gear shaper cutters design.
3. Facilitate design process and reduce product design time.
4. Develop a solid software solution for the design of different kinds of gear cutters.
5. Increase geometrical calculations precision.
6. Enable accumulating and re-using knowledge about the designed products.
7. Ensure possibility to export technical drawings in a format, compatible with other popular CAD packages.
8. Ensure confidentiality and security when using the system.
9. Deliver well documented source code together with the finished software for the further development and maintenance.

Figure 6.1 presents different considerations related to the Web-based CAD/CAM system for GSCs and highlights common decision points in the development process, that need to be kept in mind. The diagram helps to picture the software product being developed taking into account usage context, domain specific considerations, enabling technologies, source code and database aspects, as well as the scope of the case study and the desired quality of the final software.

6.2.2 Business processes and workflow

Gear shaper cutter design process can be divided in several interrelated subprocesses, which include input of cutter or master gear details, calculating rack and cutter profiles, preliminary inspection of tool characteristics, calculating conjugate gear profile, checking if the gear tooth profile is within the specified tolerances, specifying tool body form and manufacturing data, as well as producing tool drawings and CNC code.

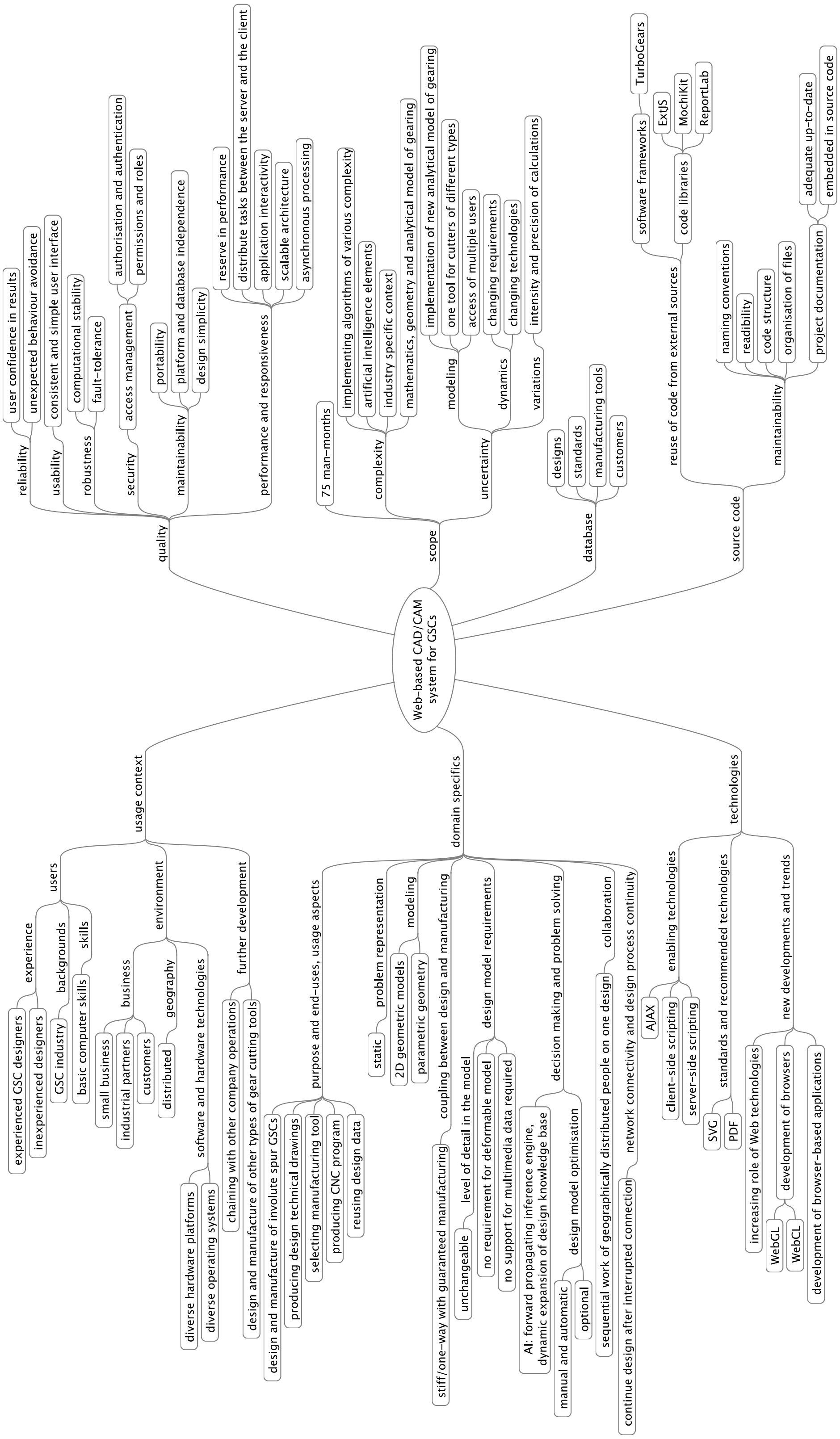


Figure 6.1: Mind-map containing different considerations related to the Web-based CAD/CAM system for GSCs

"Gear shaper cutter design could be done by applying one of three different workflows each with different set of inputs and slightly different processing. The cutter only workflow allows to input the cutter parameters directly. In contrast, the gear and cutter workflow defines the cutter through the gear it should cut. Finally, the redesign workflow uses the same gear data as the gear and cutter case, but this time the gear is redefined at a different pitch diameter." [20]

The three workflows are shown in the process model of gear shaper cutter design provided in Figure 6.2 [148]. The process model comprises green blocks referring to data collection and processing, pink blocks referring to geometry processing, a blue block referring to redesign process, and visualisation and decision points marked with yellow [20].

The old gear shaper cutter design process employed at the company is based on using different software tools intended for the design of spur or helical gear shaper cutters. Printed copies of tool design data and technical drawings are used in the company's document workflow for different purposes. For example, former drawings and job tickets with manufacturing instructions are used in the shop-floor; tool drawings and printouts of design details are used in communications with customers.

6.3 Eliciting user requirements

The sources of problems faced during elicitation of user requirements for the Web-based CAD/CAM system could be categorised as follows:

- Different expertise areas of system users and developers, which leads to user requirements ambiguity.
- Uncertain and conflicting user requirements, aggravated by functional complexity.
- Insufficient awareness of users about their requirements ("unknown knowns" according to the classification, introduced by Former United States Secretary of Defence Donald Rumsfeld [152]).
- Change of users' opinion about the previously specified requirements during the software development.

In order to address these problems, user requirements elicitation is conducted in an organised manner. The case study development started with gaining as clear understanding about the scope of the case study and desired software product as possible. For

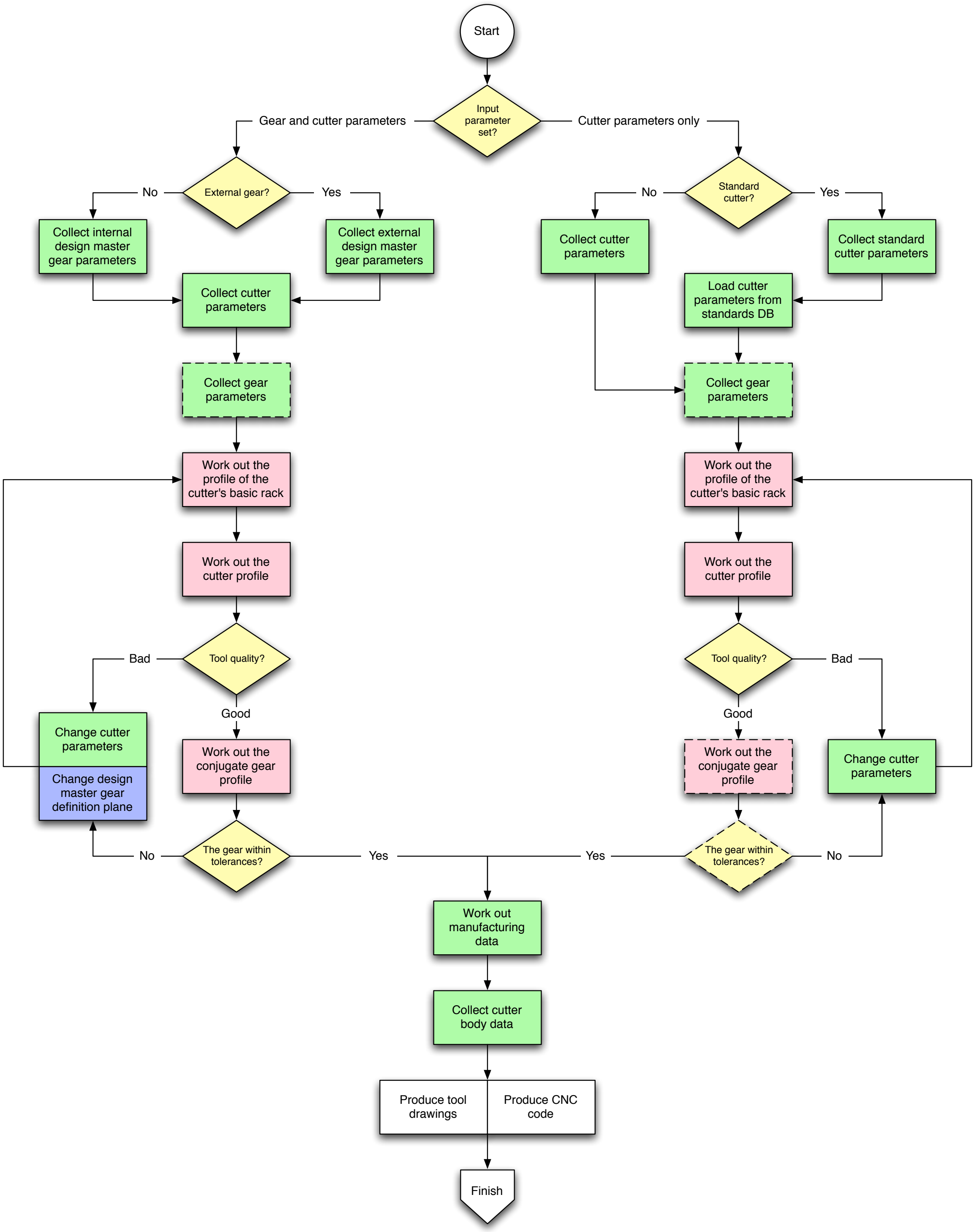


Figure 6.2: Gear shaper cutter design process [148]

this purpose the software concept is described, key use cases are identified and initial requirements are obtained. Sessions for transferring relevant industrial knowledge to the development teams have been organised to ensure that developers and stakeholders speak the same language and the developers understand user requirements properly, which leads to increased development productivity and better quality of the final software. The sessions include training in gear shaper cutter design methods, communicating definitions of terms, learning underlying principles and technologies relevant to the industry.

The methods used to elicit requirements for Web-based CAD/CAM system for GSCs include user interviews with stakeholders, prototyping and obtaining feedback by demonstrations and conducting workshops. The requirement elicitation activities were focused on the following aspects of the new software system:

- *CAD/CAM software user interface*: implementation and look; different kinds of gear cutters user interface sections; user roles and access management; control elements layout; usability and navigation; help.
- *Data input*: input fields layout and display; measurement units; predefined input values for standard tools.
- *Saving and loading design data*: design identifier (name, reference); data storage; data saving and retrieval procedures; data synchronisation; draft autosave option; collisions and data corruption.
- *Calculations*: calculated results; required precision.
- *Models and technical drawings*: implementation of technical drawings; how the drawing should look and what kind of information should be depicted on the drawing; required precision; required level of details; scale; measurement units; validation; tools for modifying and manipulating design models and technical drawings.
- *CNC code*: CNC code generation, storage and editing; machine-specific requirements.
- *Export of graphical drawings*: supported graphics formats; the look and content of exported files.
- *Display of design results*: display of design results (calculated results, model and technical drawings, CNC code) on the screen and in the exported files; printing design results.

User interviews are conducted with those users, that can provide the most complete and precise information about every aspect of the system, it is with product designers, who are the potential users of the new system; shop-floor workers, who can describe nuances in the computer-aided manufacturing; as well as managers and people from customer relations department. As user requirements can be added, refined or changed during the development, requirement elicitation activities are conducted continuously throughout the course of the case study.

6.4 Planning the development

In accordance with the methodology proposed in Chapter 5, the development of the first case study is organised in three sequential phases: initial phase, which incorporates software concept, risks assessment and initial requirements elicitation; software design optimisation and implementation of throwaway prototypes; and incremental development, divided in 18 iterations. The last phase incorporates search for solutions when required. The process model for developing the Web-based CAD/CAM system for GSCs is presented in Figure 6.3.

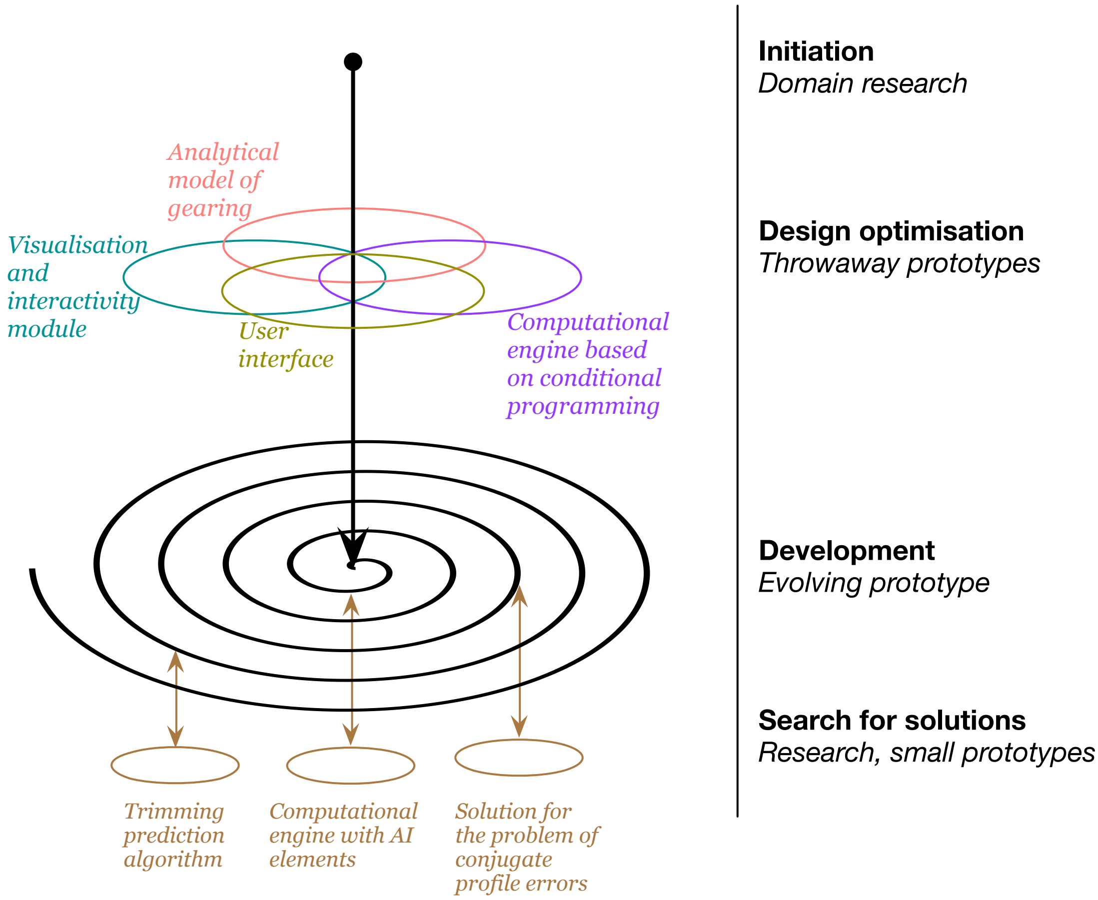


Figure 6.3: Process model for developing the Web-based CAD/CAM system for GSCs

#	Title	# Successors	# Predecessors
0	**Web-based CAD/CAM system for GSC**		
1	Project initiation	2	
2	**Software concept**	**6**	**1**
6	**Risks assessment**	**10**	**2**
10	**Initial requirements elicitation**	**13**	**6**
13	**Architectural decisions**	**27**	**10**
27	High-level structured design documentation	28	13
28	**Development planning**	**33**	**27**
33	**Incremental development**	**211**	**28**
34	**Iteration 1 – Back end**	**43**	
43	**Iteration 2 – Front end**	**52**	**34**
52	**Iteration 3 – Analytical model**	**61**	**43**
61	**Iteration 4 – Computational engine**	**71**	**52**
71	**Iteration 5 – External gear cutter design**	**81**	**61**
81	**Iteration 6 – Internal gear cutter design**	**91**	**71**
91	**Iteration 7 – Design process 1: cutter details**	**101**	**81**
101	**Iteration 8 – Design process 2: gear details**	**111**	**91**
111	**Iteration 9 – Design process 3: tool redesign**	**121**	**101**
121	**Iteration 10 – Common graph**	**131**	**111**
131	**Iteration 11 – Multigraph**	**141**	**121**
141	**Iteration 12 – Tool life**	**151**	**131**
151	**Iteration 13 – Tool body design**	**161**	**141**
161	**Iteration 14 – Manufacturing section**	**171**	**151**
171	**Iteration 15 – PDF export**	**181**	**161**
181	**Iteration 16 – CNC support**	**191**	**171**
191	**Iteration 17 – User access management**	**201**	**181**
201	**Iteration 18 – Trimming prediction for internal gears**		**191**
211	**Documentation**	**215**	**33**
215	**Testing**	**218**	**211**
218	**Roll Out**	**223**	**215**
223	Launch finished		218

Figure 6.4: Work breakdown for the development of the Web-based CAD/CAM system for GSCs

During the second phase throwaway prototypes of analytical model, computational engine, user interface and visualisation and interactivity module have been developed to find out the optimal design for the Web-based CAD/CAM system for GSCs. During the incremental development phase each iteration followed through refining requirements, updating design documentation, assessing risks, prioritising tasks, short-term planning and adjusting time and effort estimation, refactoring, implementation, testing and fixing discovered errors, integration with the production system and obtaining feedback. Search for solutions was performed several times during this phase to investigate appropriate solutions for encountered problems, such as the problem of conjugate profile errors (see section 6.9.5), design of computational engine with AI elements (see section 6.9.2) and implementation of an accurate trimming prediction algorithm (see Figure 6.15, Figure C.1 and section 6.9.1).

Table 6.1: Prioritising development tasks for the Web-based CAD/CAM system for GSCs

Iteration	**Core tasks**	**Major tasks**	**Optional tasks**
1. Back end	Create Database (DB) structure, define data types, physical data organisation, database processing programs, build connections to the front end		
2. Front end	Gear inputs; Cutter inputs; On-screen presentation of design results; Main menu; Design process control elements; Design save and load dialogue; Gear list dialogue; Tool body inputs; Manufacturing cutter and milling tool selection; Tool face inputs.	Product model manipulation controls; Inspection controls.	Circular thickness calculator; Inputs for additional design information.
3. Analytical model	Implement an analytical model of gearing for GSC design. Obtain cutter tooth profile from specified rack details. Obtain gear tooth profile from the cutter tooth profile. Algorithm for finding line segments intersections.	Solve tooth profile error problem in designs with undercuts.	Improve the performance and precision of tooth profile models. Alternative algorithms for finding line segments intersections.
4. Computational engine	Implement forward propagating inference engine. Add formulae for external and internal involute spur GSC design calculations and redesign case.		Use a general purpose logic programming language for the rule base implementation.
5. External gear cutter design	Inputs associated with external gear cutter design. Define conditions for respective inputs display. Multiple gears support in one design.	Add standard cutters specifications. Measurement units switch. In-field formula calculations for gear and cutter inputs.	Add smart input tips validation.

For estimation of effort required for the development of the Web-based CAD/CAM software for GSCs judgement-based approach has been utilised, as it usually is more accurate than model-based effort estimate in situations, where important domain knowledge, high level of change and estimation uncertainty cannot be included in the estimation models [153, 154]. The guidelines for applying the expert estimation process are well described in the literature [155].

Considering the expert judgement of the development team about the available development technologies, available domain expertise and the novel software development approach, the estimated effort for the case study development is about 75 man-months, which translates into 25 months of required time with a team of 3 developers, combining expertise in software development, GSCs design and manufacture methods and techniques and analytical gearing models. The work breakdown for the Web-based CAD/CAM system development case study is presented in Figure 6.4 and has been used to aid the effort estimation process.

As practice shows that average overruns in software development projects are about 30% [156], the reserve threshold for the Web-based CAD/CAM system for GSCs development case study has been chosen $m = 30\%$. The lag δ is set to 3% to avoid excessive fluctuations in the speed of reserve utilisation.

Table 6.1 provides a few examples of prioritising development tasks for each iteration. In accordance with the proposed methodology, tasks are divided in three categories: core, major and optional tasks (see section 5.3).

Table 6.2: Time reserve management for the development of the Web-based CAD/CAM system for GSCs

Iteration	**Planned and reserved time for the rest off the development,** *days*	**Current reserve** c**, %**	**Scheduled tasks**
1. Back end	Planned time = 550; Reserved time = 165	30	Core + Major
2. Front end	Planned time = 530; Reserved time = 165	31	Core + Major
3. Analytical model	Planned time = 475; Reserved time = 160	34	Core + Major + Optional
4. Computational engine	Planned time = 460; Reserved time = 120	26	Core
5. External gear cutter design	Planned time = 430; Reserved time = 130	30	Core + Major

Table 6.2 demonstrates on a few iterations, how the time reserve is managed. Assume, the average number of working days per month is 22, then planned case study duration is 550 days and reserved time is 165 days.

As it follows from the provided data, the reserve was utilised too slowly during the first two iterations. As a result, at the beginning of the third iteration current reserve c exceeded the 10% tolerance of the reserve threshold. This allowed to schedule for the third iteration implementation of optional tasks in addition to core and major tasks. In turn, this reduced the reserve to 26%, which is lower than the bottom tolerance $m - \delta = 27\%$, therefore the tasks, scheduled for the fourth iteration include only tasks with highest priority. Subsequently, at the beginning of the fifth iteration the reserve is normalised to the threshold of 30% again.

6.5 Resource allocation and collaboration

The development team for the Web-based CAD/CAM system for GSCs is created specifically for this case study and involves people with expertise in software development, manufacturing of gear cutting tools and the area of CAD/CAM. Also short-term involvement of external experts in gearing geometry and mathematical algorithms was required for solving problems faced during the development. Intensity of engagement of different stakeholders in the Web-based CAD/CAM development is based on their commitment and influence.

Table 6.3: Engagement of different stakeholders in Web-based CAD/CAM development

Involvement	**Onsite collaboration**	**Online collaboration or teleconference**
Day-to-day development	Developers and experts in GSC manufacturing, involved in the development. Participation of CAD/CAM experts as required	CAD/CAM experts
Weekly progress review	Developers and experts in GSC manufacturing, involved in the development	CAD/CAM experts, executive manager and facilitator
Monthly progress review		Key stakeholders
Quarterly meeting	Key stakeholders	
Temporary participation	Online or onsite collaboration of external experts with the development team as required for solving problems, faced during the development	

Table 6.3 defines the involvement of different stakeholders in the case study development activities, considering the intensity of engagement and collaboration. It is worth to mention specifically, that the role of experts in GSC manufacturing, involved in the development, varies as the development progresses, shifting the focus from participation in requirements elicitation and knowledge base construction towards outcome evaluation, validation and verification.

Task prioritising and flexible project schedule enabled resolving of stakeholder availability challenges and ensuring good productivity of the development team by defining optimal level of stakeholder engagement using onsite or online collaboration capabilities.

6.6 Creating optimal design of the software

During the design of the Web-based CAD/CAM system for GSCs, the most critical decisions regarding the application interactivity, task distribution between the server and the client and multiuser interaction have been made considering the business requirements and desired overall performance of the system. Figure 6.5 illustrates design decisions defining interactive capabilities of the Web-based CAD/CAM system, in particular handling application events with scripting languages, using synchronous and asynchronous requests for effective management of data inputs and outputs, implementation of graphics and mechanisms for manipulations with it using vector graphics and scripts, as well as optimising system responsiveness to fit within the performance and latency constraints and ensure real-time design results.

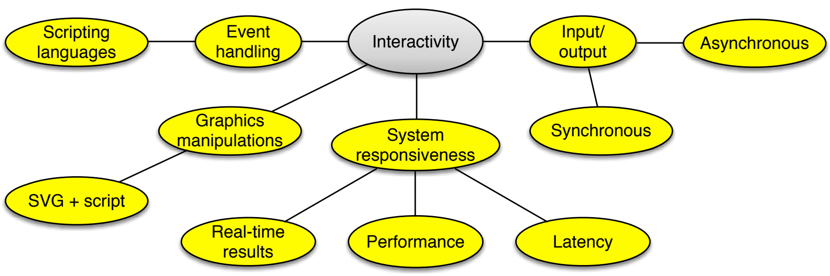


Figure 6.5: Interactivity related design decisions for the Web-based CAD/CAM system for GSCs

As the performance of a Web-based CAD/CAM application is greatly affected by how tasks are distributed between the client and the server, relevant design decisions are critical for creating optimal application design. Major application tasks designed to be executed on the server and on the client in the Web-based CAD/CAM system for GSCs are listed in Table 6.4. The operations directly related with interactions between the user and the system are mostly performed on the client side to minimise client server interactions and improve system responsiveness. Subsequently computation-intensive tasks are implemented on the server side of the Web-based CAD/CAM system.

Table 6.4: Design decisions about task distribution between the server and the client in the Web-based CAD/CAM system for GSCs

Server	**Client**
Design data calculations; Output file generation; Generating CNC programs; Optimisation.	Manipulations with graphics; Data input; Input data validation; Display results.

Design decisions regarding collaboration features of the Web-based CAD/CAM system for GSCs define mechanisms for sharing resources and knowledge and interaction in different time/same place and different time/different place, without capabilities for simultaneous work on a task. Collaboration related design decisions are marked in Figure 6.6.

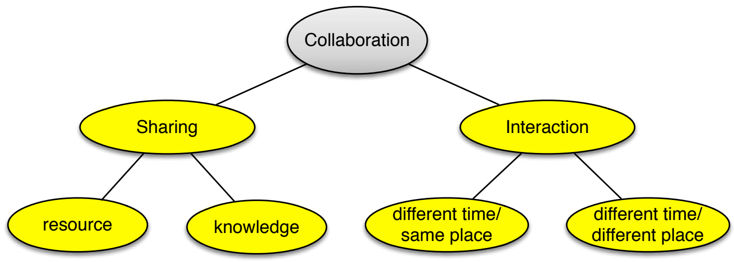


Figure 6.6: Collaboration related design decisions for the Web-based CAD/CAM system for GSCs

By addressing the critical design decision points the most appropriate technologies and application infrastructure are investigated and application design is further optimised using throwaway prototypes. Then throwaway prototypes are developed for the most critical parts of the Web-based CAD/CAM system to evaluate the initial design and easily make changes if needed. For the described case study of the Web-based CAD/CAM system for GSCs the most critical parts of the design include analytical gearing model, computational engine, user interface and visualisation and interactivity module. Throwaway prototyping of these parts of the application design covers some of the software product considerations related to the domain specific considerations, project scope, application usage context, software quality and implemented technologies, which are mind-mapped in Figure 6.1 and marked in Figure 6.7 with colours to show the coverage by the corresponding prototypes. Figure 6.3 shows how the development of throwaway prototypes for analytical gearing model, computational engine, user interface and visualisation and interactivity module is in line with the case study development process model.

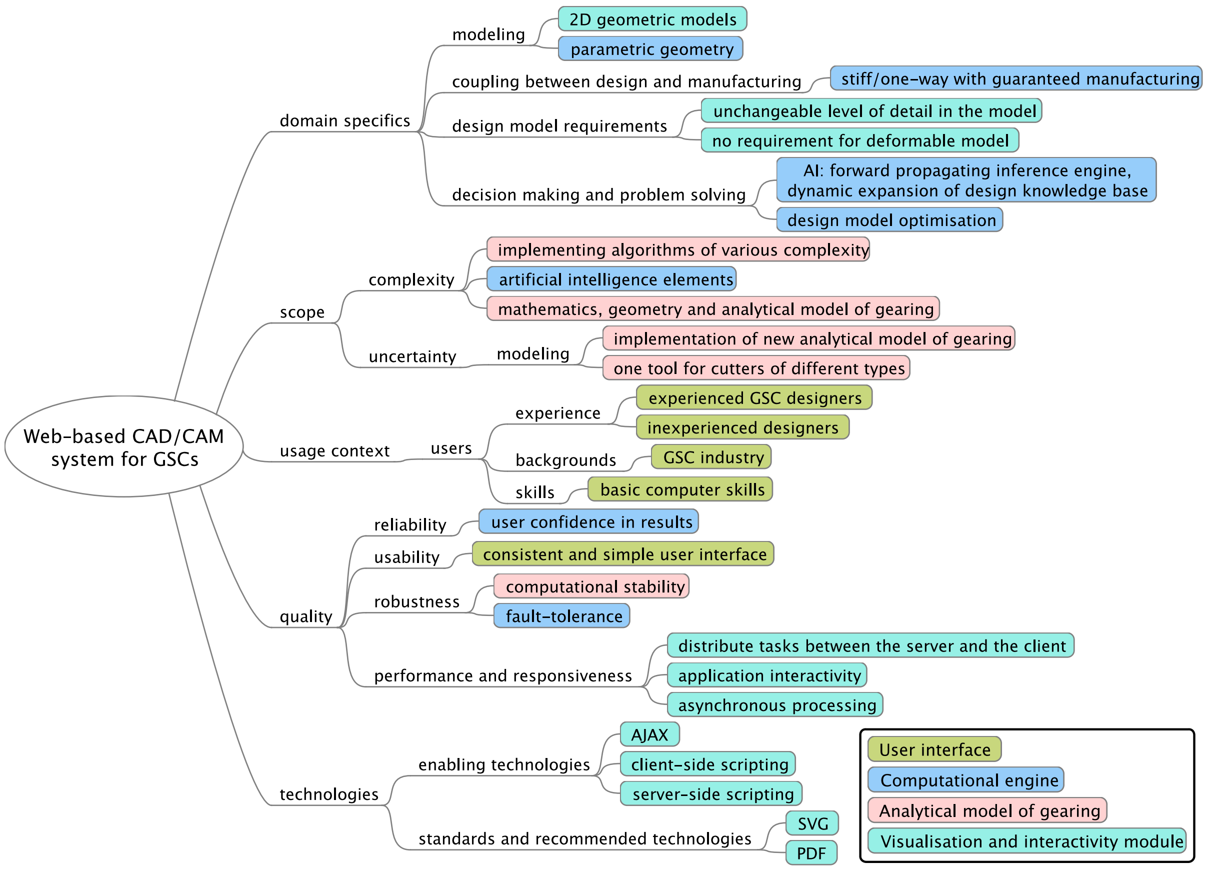


Figure 6.7: Addressing design considerations for the Web-based CAD/CAM system for GSCs using prototypes for user interface, computational engine, analytical model of gearing and visualisation and interactivity module

Conceptual representation of the subject area is shown in Figure 6.8 with the use of Entity-Relationship Model (ERM). Every single design could encompass up to six entities: one or multiple gears, which are intended to be cut by a particular designed cutter; cutter itself; cutter body shape; teeth pattern (missed teeth and bold spaces); a tool used to produce designed gear shaper cutter and basic rack. Design data could be saved in the database and reused later. Upon the design completion generated CNC code is available for a shop-floor worker to produce the final product.

The use case diagram of Web-based CAD/CAM system for gear shaper cutters is shown in Figure 6.9 and describes interactions between the user and the system. Only users registered in the database can access the Web-based CAD/CAM system and gear

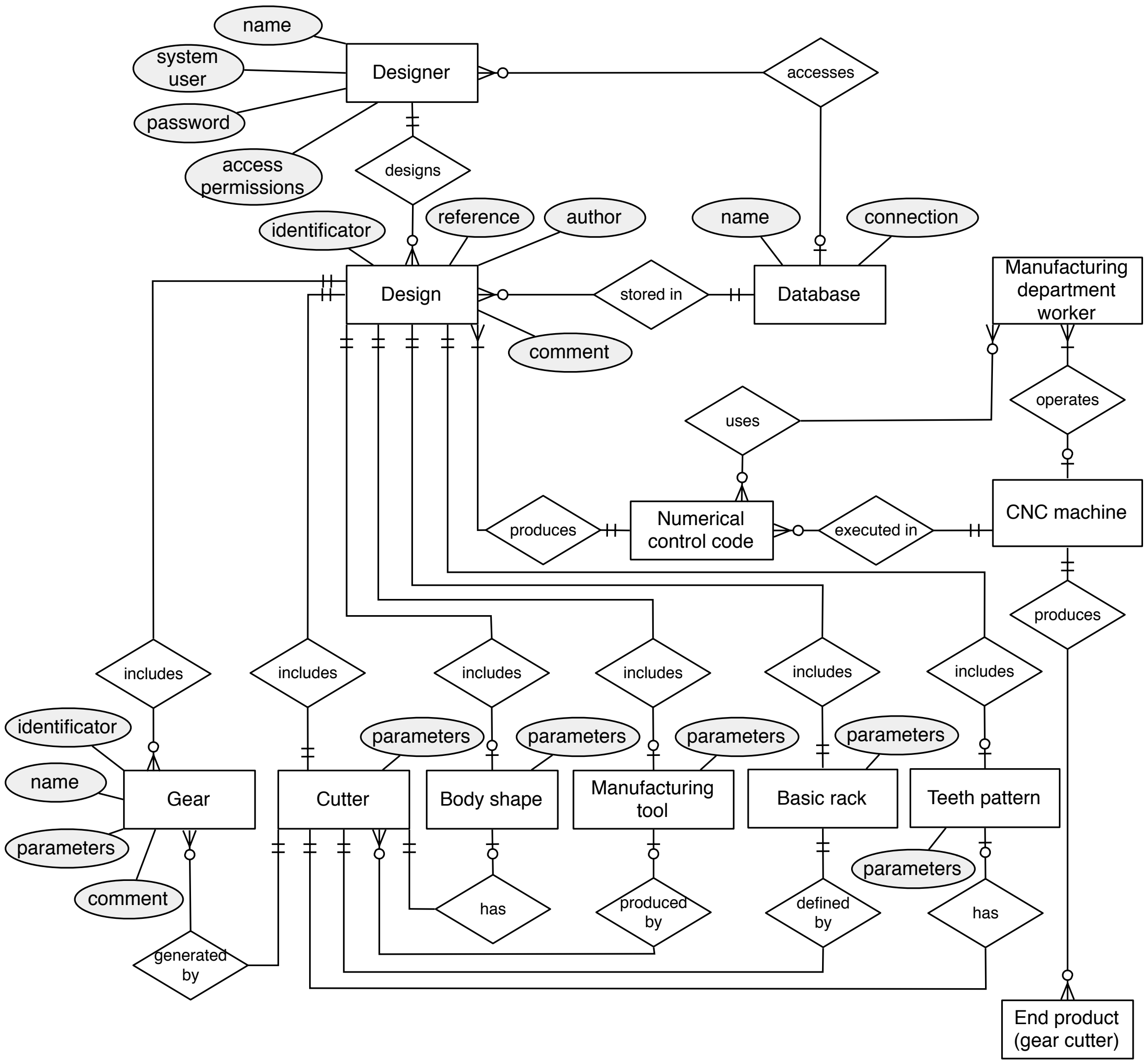


Figure 6.8: Entity-relationship diagram of Web-based CAD/CAM system for gear shaper cutters

shaper cutter design database. There are three groups of users defined in the system, who have different roles in the system and use it for different purposes. Customer Relationship Management (CRM) employees are only able to view designs that are stored in the database and fill in initial design data, designers in addition to this perform all operations associated with gear shaper cutter design and administrators are responsible for user permission management.

Industrial CAD/CAM systems usually are large applications with complex business logic and consisting of a big number of different components. In order to handle all the components of Web-based CAD/CAM application and reconstruct the whole page effectively each time it is changed an architectural pattern called Model-View-Controller

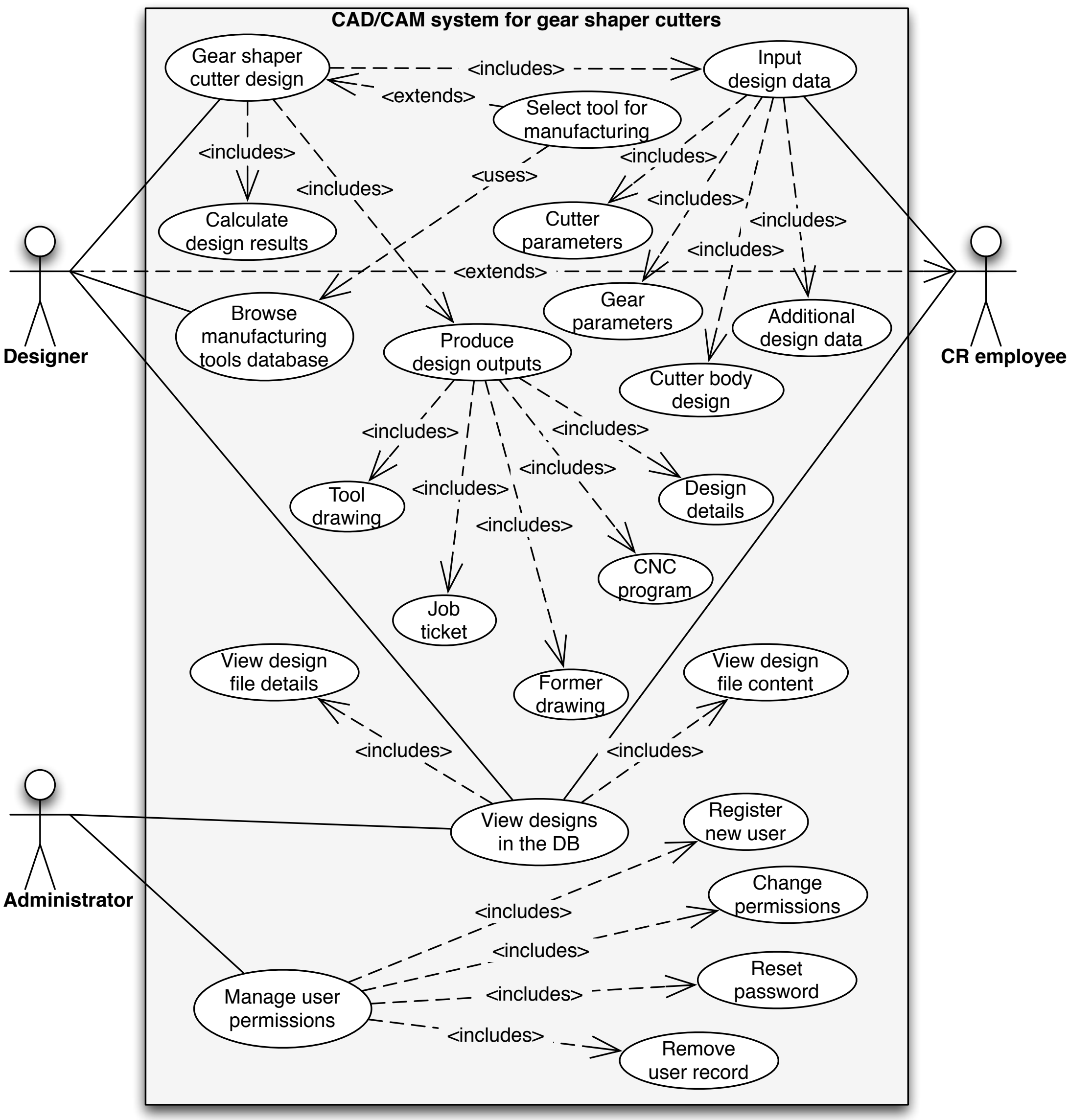


Figure 6.9: Use case diagram of Web-based CAD/CAM system for gear shaper cutters

(MVC) is applied. The pattern allows to separate business logic from content and presentation, enabling to develop, test and maintain each part independently from others. This results in better flexibility of program code in the development process. Moreover, model isolated from view is more robust and unlikely to be corrupted by the developer while modifying the view.

Desktop-like responsiveness and interactivity of the Web-based CAD/CAM application is ensured by utilising AJAX technology for handling user interface events at specific UI components on the page.

"The user interface is developed to maximise CAD system usability and support designers with different level of technical knowledge in the relevant field. To ensure the displayed information can be effectively managed, the interface implements collapsible panels, therefore only necessary amount of controls gets displayed each time." [20]

Using WWW consortium recommended [28] vector graphic format SVG enables visualising 2D design data directly in the Web browser, without the use of supplementary programs. This way platform and browser independent of the Web-based CAD/CAM system is enabled by means of freely available components and standards [20].

6.7 Software architecture and implemented technologies

"The key functional requirements for gear shaper cutter CAD package are the ability to produce cutter and gear profiles from the input data, visualise them, assist design refinement and, finally, produce tool drawings and outputs suitable for CAM system. The system should be future proof, in terms of modular composition, scalability and open interconnection with other infrastructure, in particular CAM and CRM systems." [20]

Gear shaper cutter design undergoes a one-way transformation of parametric input data to a graphical model of conjugate cutter and gear profiles and design and manufacturing details. As direct modification of the graphical models is not required, a browser-based implementation will be capable to handle the CAD/CAM system for GSC. [20]

Software system architecture incorporates the set of determining decisions about its organisation, including structural elements, their interfaces, behaviour and composition in larger subsystems, as well as an architectural style that guides this organisation [157].

"Referring to Figure 6.10 the client portion of the software is based on the templates **C** written in a purpose-built language that are being served by the server-side controller **B** in response to the client request **A**. Textual inputs **F** are being transformed to design-time results **G**, namely, visual representation of the conjugate profiles and accompanying design parameters by means of client-side **I** and server-side **B** controllers. The visuals use the SVG format for the graphical data interchange and representation." [20]

"The interactive features of the user interface are supported by MochiKit and ExtJS JavaScript library that provides an abstraction layer from the browser-specific implementation of the JavaScript runtime. A desktop-like experience when using the software is implemented using the AJAX technology that uses JavaScript Object Notation (JSON)

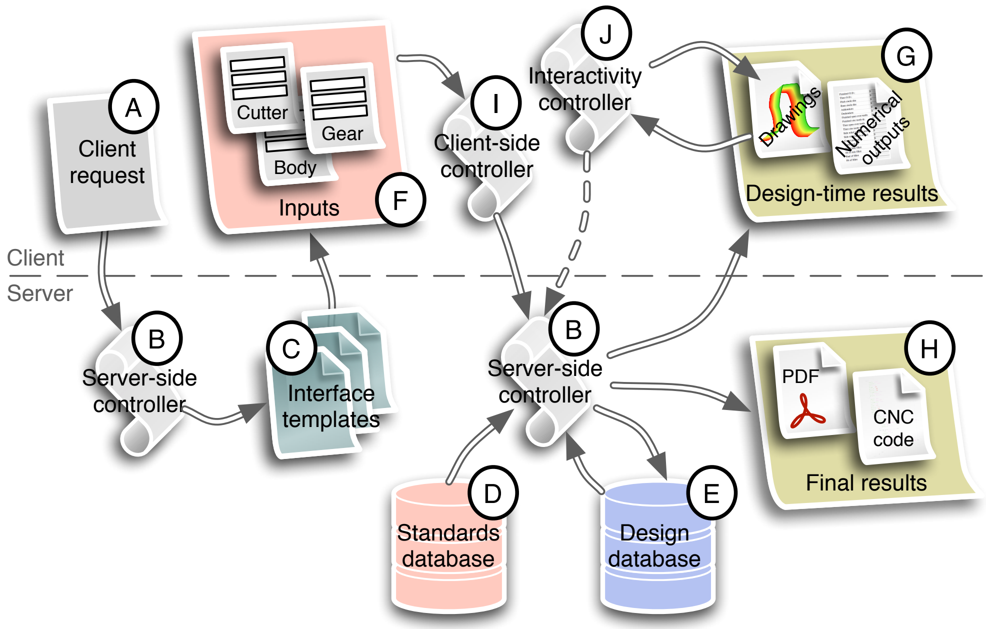


Figure 6.10: Architecture of Web-based CAD/CAM system for gear shaper cutters

as the data interchange format between the client and the server, so that data exchange between the client and the server portions of the software takes place without reloading the web page. To speed up the design process, both the database of existing designs **E** and the standards database **D** are integrated with the system." [20]

Server portion of the system is implemented using Python programming language and TurboGears framework. Python has been chosen for the implementation of controllers, because it enables developing software solutions of different level of complexity within short time-frames, while supporting various programming paradigms (e.g. functional, object-oriented and imperative programming). Programming code written in Python is concise and easy readable, that in turn increases developer's productivity and lowers maintenance costs. TurboGears framework facilitates processing of Web-queries and ensures authentication services and run-time environment.

Database management system PostgreSQL has been chosen for the development of design database for its safety and performance.

The Web-based CAD/CAM system is entirely based on Web technologies and is compatible with different program environments and platforms. The system can be installed on the server running Microsoft Windows, GNU/Linux or OS X, and accessible from anywhere through the Internet using a modern Web browser. The system does not require any additional components installation on the client side for enabling full functionality. Table 6.5 describes deployment requirements for the Web-based CAD/CAM system for gear shaper cutters, which are derived mainly from the design of the system as the customer hardware constraints fall within these deployment requirements.

The development of the system is realised on OS X 10.4 - 10.7 platform, using JEdit and Eclipse Integrated Development Environments (IDE) and such tools as FireBug, WebKit and Drosera. Supporting the idea of open source software, preference in the application implementation has been given to GNU/Linux (Ubuntu 8.40 - 10.04) operating system for the installation of the system working copy.

Table 6.5: Deployment requirements for the Web-based CAD/CAM system for GSCs

	Server	**Client**
Operational environment	• Microsoft Windows starting with Windows 2000, Mac OS X 10.3 or later, Linux starting with core version 2.4.12; • PostgreSQL 8.0 or later; • Python 2.4 (or later), TurboGears 1.1, Python Imaging Library (PIL) 1.1.6 (or later), Reportlab 2.2 (or later), Python module RBTree 1.6 (or later); • Backup software and hardware.	• Operating system able to run Safari 3.0 (or later), FireFox 2.0 (or later) or Google Chrome Web browser; • Safari 3.0 (or later), FireFox 2.0 (or later) or Google Chrome Web browser.
Hardware	• IA-32 or AMD64 architecture (compatible with architectures MIPS, POWER, PowerPC and SPARC); • Central Processing Unit (CPU) 1 GHz (3 GHz dual core system recommended); • RAM 1 GB (3 GB recommended); • Redundant Array of Independent Disks (RAID) 1 300 GB; • Gigabit Ethernet.	• Any computer and operating system able to run Safari 3.0 (or later), FireFox 2.0 (or later) or Google Chrome web browser.
Communication requirements	• Internet connection starting with 1 Mbit (data rate might be higher for increased number of Web-based CAD/CAM system users).	

Modern programming technologies offer the user of Web-applications functionality similar to desktop applications. This includes features traditionally are associated with desktop applications, such as collapsible side panels with design inputs and numerical outputs, interactive drawings and mathematical formulae transformation to numerical result inside input field.

Starting design of a gear shaper cutter, user inputs parameters, which define cutter geometry. Client-side controller **I** checks if user inputs **F** are correct and sufficient. Client-side controller allows entering mathematical formulae instead of input parameters and converts them to calculated values.

After the data have been entered in the system, they are sent to the server-side controller **B**, which subsequently performs necessary calculations, produces cutter and gear tooth profile drawings and returns them back to the browser together with calculated numerical outputs **G**. "As gear shaper cutters design is described through large number of parameters and requires extensive computations, forward propagating inference engine has been implemented" to ensure that new calculated results expand system's dynamic knowledge base [20]. "The engine uses elements of artificial intelligence and includes unification procedure of current state and conditions and result evaluation procedure" [20].

Interactivity controller **J** enables real-time information updates and performing of such actions, like drawing zoom in and zoom out, dragging the drawing inside the view area, measuring of radii and distances. Designer can change cutter parameters and experiment with different tooth geometry modifications.

As the result of design process the CAD/CAM offers to create technical tool drawings in Portable Document Format (PDF) files and CNC code, which could be modified later in the CAM module. Both PDF documents and CNC code generation functions are implemented using the server-side controller.

Design database **E** and standard database **D** allow design data reuse. In the case of standard cutter design server-side controller obtains cutter parameters from the standard database.

6.8 User interface

The new Web-based CAD/CAM system for gear shaper cutters operates with large amounts of design information and technical drawings. Therefore, to fit all important design information and graphs simultaneously in one screen the main program screen contains small graph previews with less details. There is an option to review detailed design information and expand each graph preview for more considerable analysis. These features are supported by modular user interface, which consists of expandable and collapsible panels containing input fields, design numerical outputs and technical drawings. Panels could be hidden or expanded independently from each other.

The Web-based CAD/CAM system for gear shaper cutters ensures high level of user interface personalisation, allowing the user to adjust size of panels, manage hidden and visible components, change column width, order and data sorting in tables, in accordance with his personal preferences.

The entire Web-based CAD/CAM system consists of three Web pages. Those are used for CAD module itself, user authentication and user profile data modifying.

User authentication and user profile screen

To be able to use the CAD/CAM system user has to be authenticated. User authentication screen contains *login* and *password* input fields. Input data has to be in text format. When the user submits entered data, the user name and password are compared with the data stored in the database. User authentication form as well as user profile data form is shown in Figure 6.11.

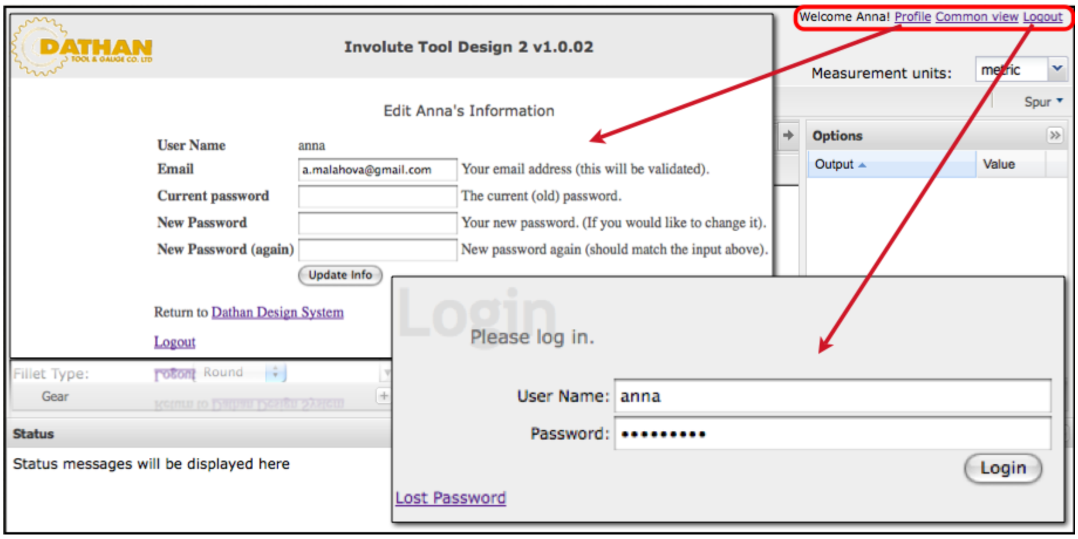


Figure 6.11: User authentication and profile data editing

User profile data form allows modifying user e-mail address and changing password. In order to change password the user is required to enter his current password, new password and then confirm the new password by retyping it second time. User profile screen contains options to return to the design system or log out.

Main screen

System main screen consists of five areas (Figure 6.12): central area, top, bottom, left and right panels. Main program menu, design measurement units switch, current design reference and options for modifying user profile data and session termination are located at the top of the main program screen.

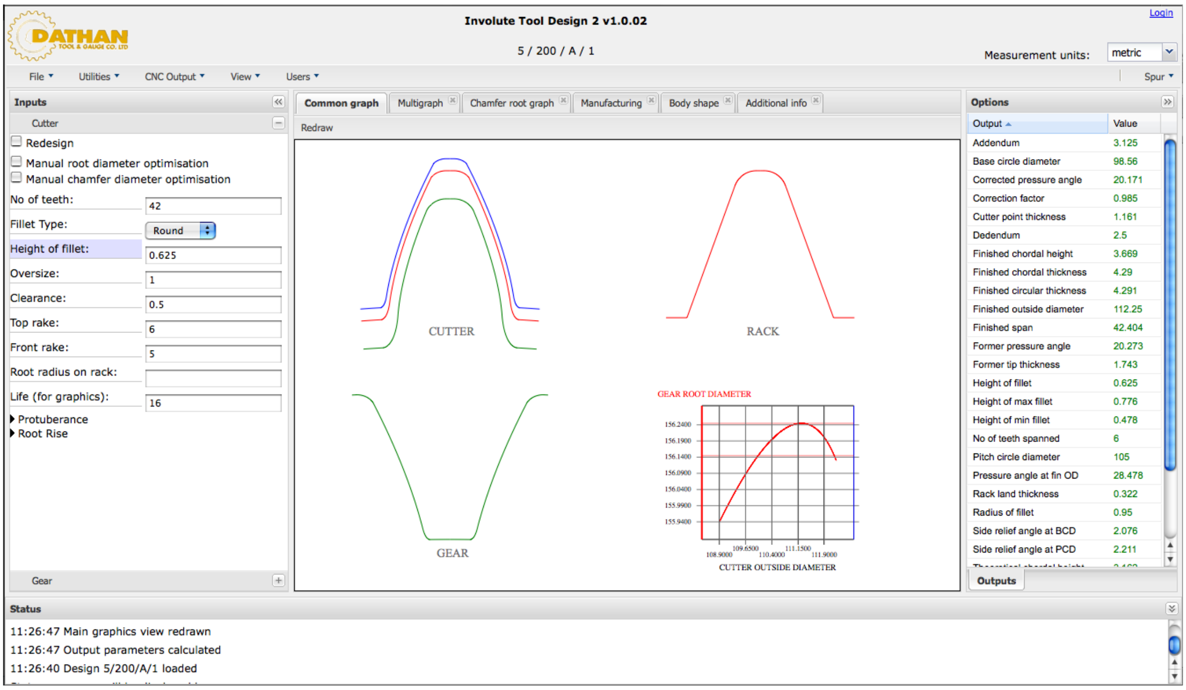
Main program menu contains five sections: File, CNC Output, View and Users (available only for users with administrative permissions) and switch between spur and helical tool design. Menu File includes the following options: New, Open, Save, Save As, Save Copy As, Export PDF, Export To File, Import From File, Delete Design (Figure 6.29). Menu CNC Output offers options for Haas and Diaform CNC code generation (Figure 6.30). Menu View has options, which allow defining of hidden and expanded tabs in the central area of the main program screen (Figure 6.18).

Bottom part of the main program screen is used for the output of system messages. The middle part of the main program screen is divided vertically in three parts. On the left side of the main program screen is a panel with input fields for cutter (Figure 6.12(a)) and gear (Figure 6.12(b)) parameters.

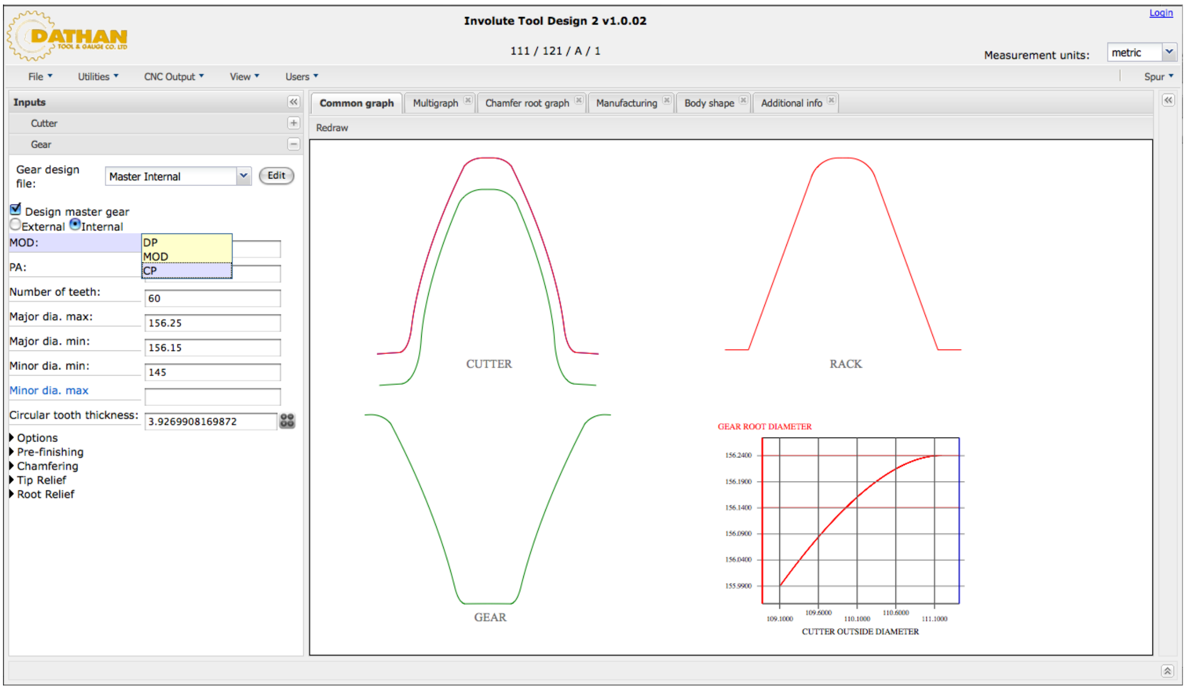
The central part of the main program screen is devised for the output of graphical information, produced by the system. The central part contains mainly graphical information and is divided in seven tabs: common design information, multigraph, gear chamfer and root diameter graph, manufacturing tool selection, body shape, tool face drawing and additional design information. The contents of the right panel depends on what is currently selected in the central area of the main program screen. When the user selects a tab in the central area of the main program screen, corresponding content is shown automatically in the right panel.

Common design information

Common design information aim is providing the user with the information, sufficient for quick estimation of obtained design results. This tab includes small previews of gear tooth profile, basic rack, cutter tooth profiles at different outside diameters which demonstrate how the profile changes throughout the tool life, as well as diagram of gear chamfer and root diameter dependency on tool life (Figure 6.12). When common design information tab is active in the central part of main program screen, the right panel contains numerical design outputs (Figure 6.12(a)).



(a) Cutter input parameters



(b) Gear input parameters

Figure 6.12: System main screen contains collapsible panels with cutter and gear inputs in the left part of the screen, graphical outputs are presented in its central part and calculated design results are listed in a collapsible panel on the right side. Common design information tab is selected in the central part of the program screen and it presents tooth profiles of the cutter, basic rack and gear, and also chamfer root graph

Multigraph

Multigraph is intended to show detailed drawings of basic rack, gear and cutter tooth profiles. When multigraph tab is active in the central part of the screen, the right panel contains multigraph options as seen on Figure 6.13.

Multigraph provides the designer with interactive tools for tooth profile inspection, such as zoom in and zoom out (Figure 6.14), measurement of radii on the tooth profile and distance from the tool centre to any selected point on the profile (Figure 6.13). The user can freely move profile radius measurement tool and change its radius using mouse scroll. Corresponding radius value is shown on the graph toolbar.

Tooth involute deviation graph shows how close the profile is following true involute line. Together with profile break points involute graph allows identifying whether different parts of tooth profile, like root relief for example, fit within the allowed tolerances or not.

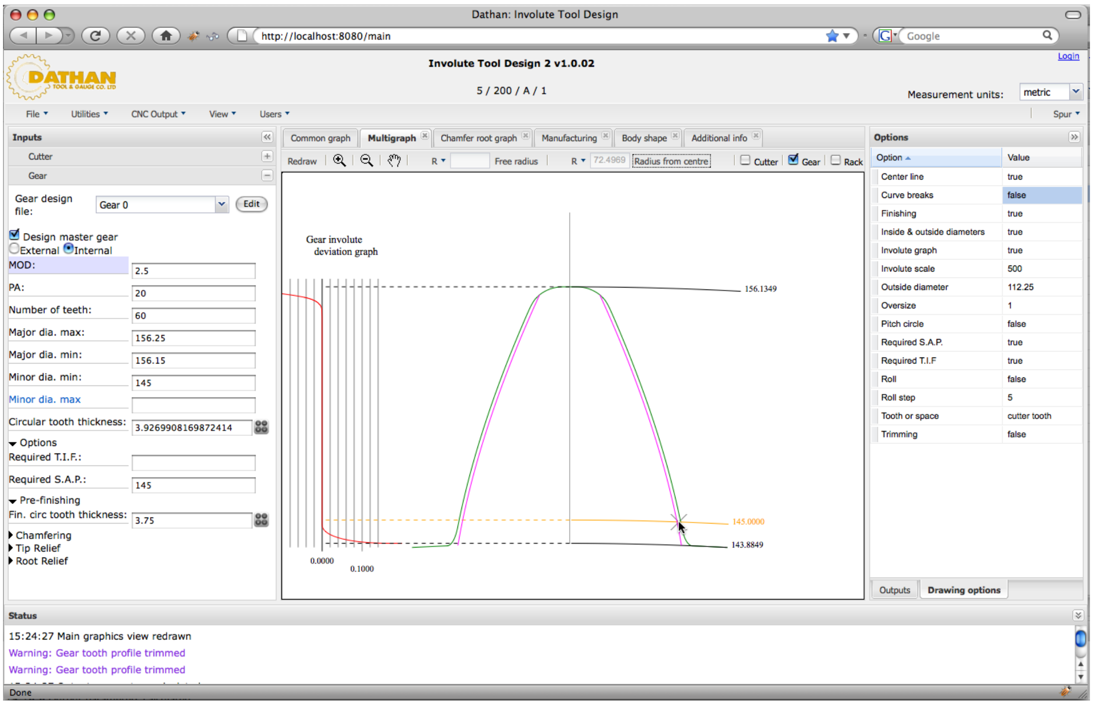
Visualisation of gear generation by the cutter (shown in Figure 6.14) or cutter generation by basic rack illustrates the path of cutting profile during the roll motion. The step for roll visualisation is adjustable and defines the level of details in the drawing.

Figure 6.15 shows visualisation of interference between internal gear and cutter profiles. Trimming prediction is an important part of internal gear cutter design, as it allows determining if a particular cutter could be used to produce required gear profile.

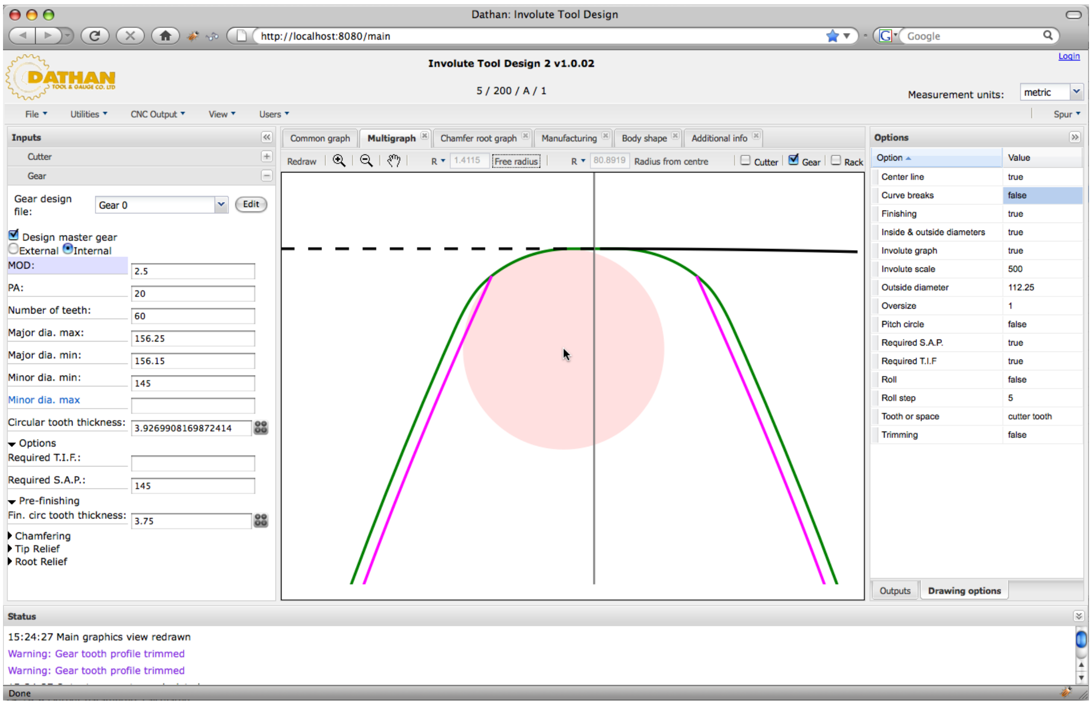
Gear chamfer and root diameter graph

Figure 6.16 presents main program screen with gear chamfer and root diameter graph section active. It illustrates variations in the gear root and chamfer diameter that occur due to changes in tooth profile of the cutter, caused by re-sharpening cycles.

To ensure that gear root and chamfer diameters are always within the given tolerances throughout the tool life, optimisation is performed by the system automatically. In some cases it might be necessary to make the optimisation manually. To support the designer in this task, minimum and maximum diameter values and tolerances are marked on the graph. The options for manual root and chamfer diameter optimisation are located in the cutter inputs panel on the left side from the graph. Numerical design outputs (similar to Figure 6.12(a)) are shown in the right panel, when the chamfer root graph tab is active in the central area.

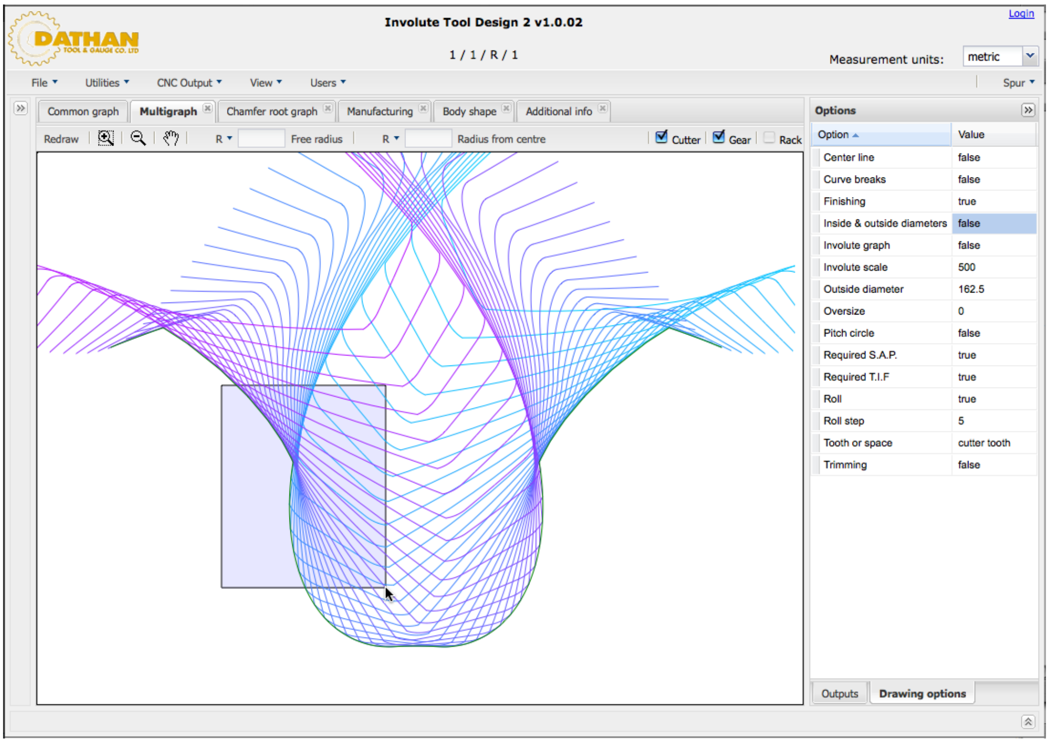


(a) Measurement of radial distance from the tool centre by dragging a small cross over the tooth profile

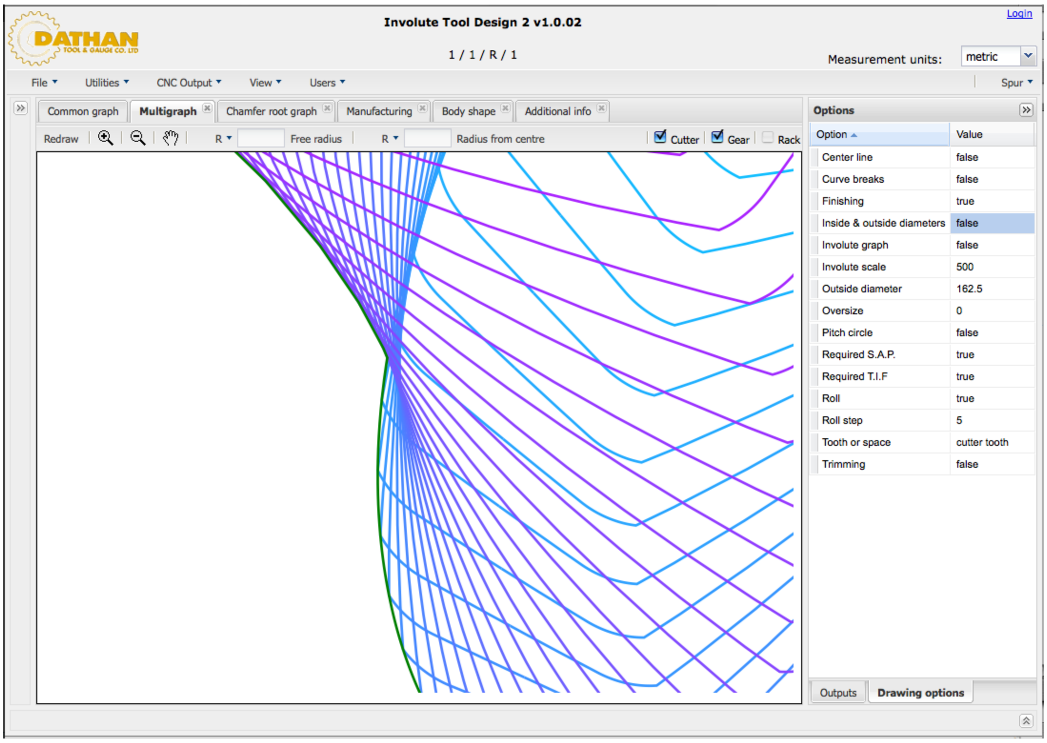


(b) Measurement of radii on the tooth profile by comparing those against a special circle (pale red) that can be freely dragged and resizing

Figure 6.13: Multigraph and tooth profile inspection tools: radius and distance measurement tools. Gear tooth profile (green) with pre-finishing (magenta) and start of active profile diameter (yellow) in (a) is under review



(a) Area selection for zoom in



(b) Enlarging the selected area

Figure 6.14: Visualisation of the roll motion when generating gear (green) with the cutter (from purple to cyan for different positions of the cutter tooth). Zoom in option is demonstrated

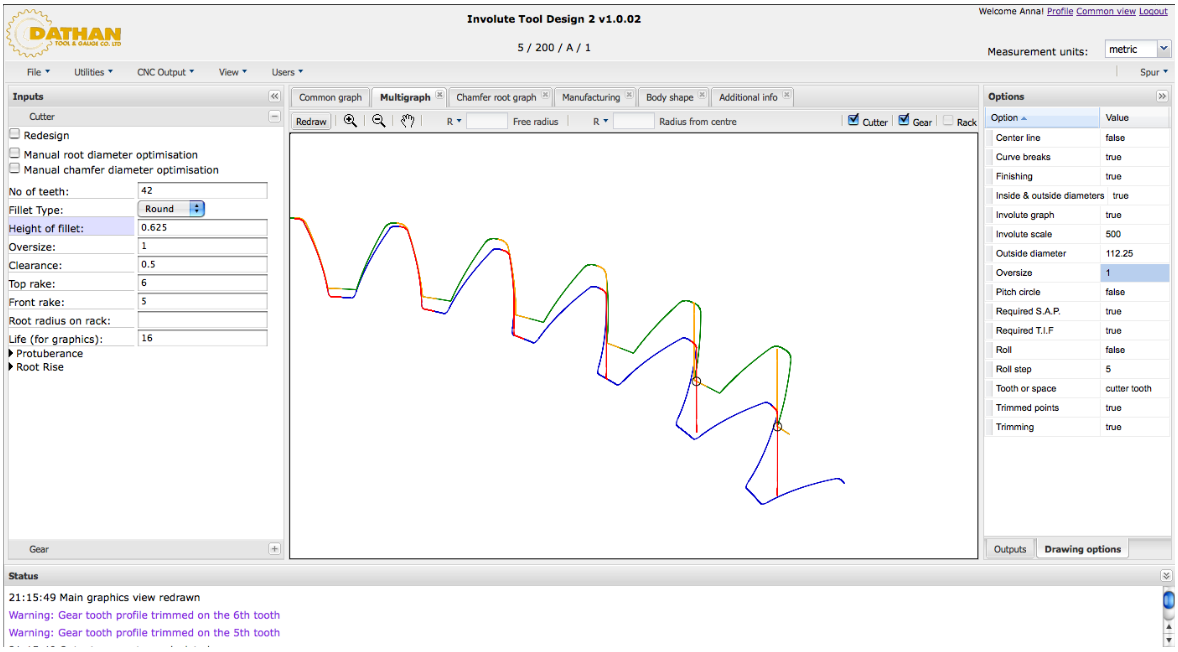


Figure 6.15: Interference prediction between internal gear (green) and cutter (blue) profiles. Red lines show the path of each cutter tooth during gear generation process (tool feed in and back off). Interference occurs when red line crosses yellow line, which shows the allowance based on gear tooth geometry

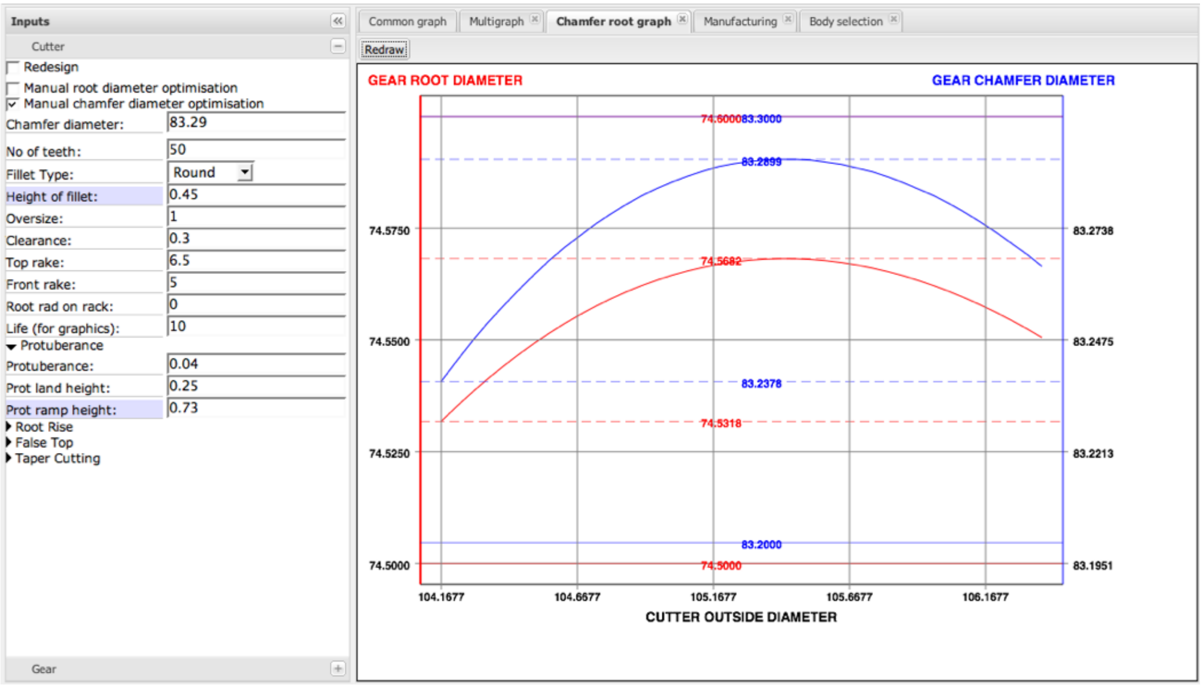


Figure 6.16: Dependency of gear chamfer and root diameter on tool life. The graphs are shown together at different scales and track changes of both diameters as the cutter wears off and its outside diameter decreases during the use. Given tolerances are depicted to guide the optimisation if required

Manufacturing tool selection

In order to decide, which manufacturing tool suits better for producing a particular cutter, the user is provided with a list of available manufacturing tools, that is shown in the right panel (Figure 6.17). Manufacturing tools with base pitch equal or close to the base pitch of designed cutter potentially could be used for producing the cutter. Those tools are highlighted with red colour in the list of all available manufacturing tools. While selecting a tool for manufacturing the cutter, user considers such parameters, as base pitch of the tool, clearance for topping, cutting depth and tool type.

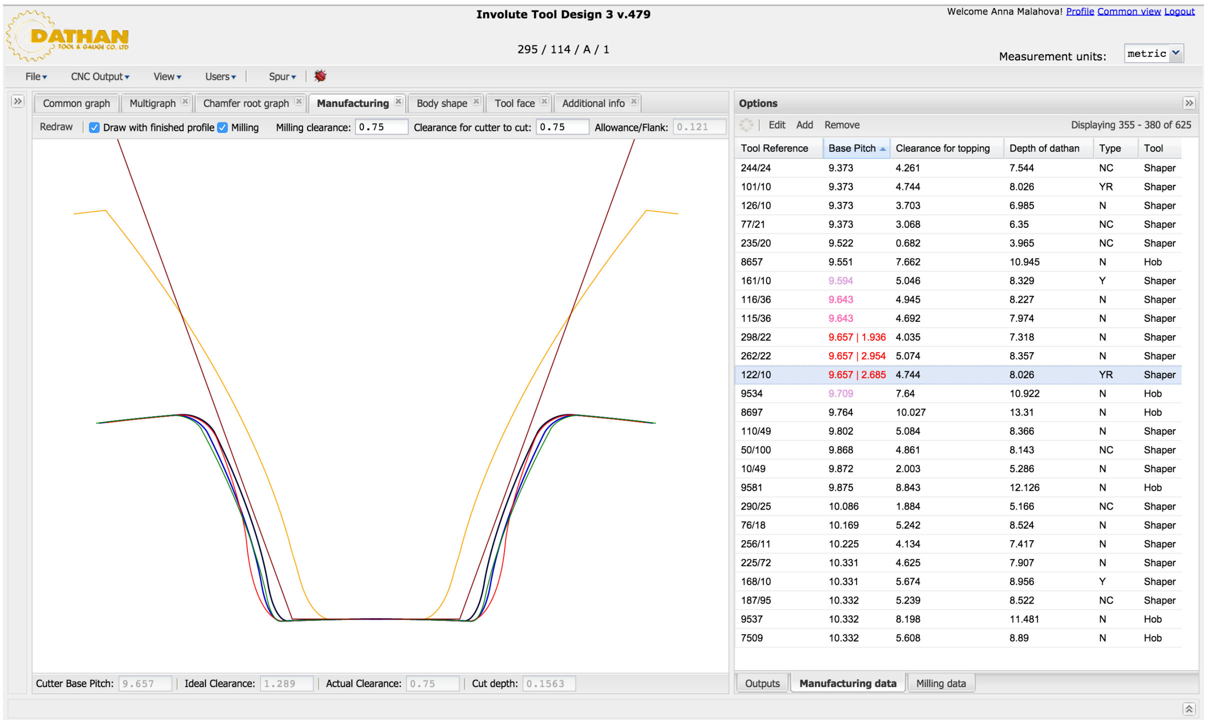


Figure 6.17: Manufacturing tool selection. Available manufacturing tools are listed in the collapsible panel on the right side from the drawing, which shows finished cutter tooth space profile (blue), cutter profile with oversize (green), cutter profile at the end of tool life (red), finished cutter profile with allowance on flanks (black), the profile cut in the cutter blank by the selected manufacturing tool (yellow) and the selected milling tool profile (brown)

The designer can visually estimate the profile, that will be cut by selected tool in the blank (red line on the graph), and decide, how it fits with the cutter tooth profile at the start (blue line) and end of its life (yellow line). Milling tool (shown in brown colour) can be also selected in this section, by picking up an appropriate milling tool from the third tab in the right panel. Both milling clearance and clearance for cutter to cut are adjustable. Finished profile can be drawn if necessary to visually estimate how the tooth shape, obtained during the manufacturing process, will fit within given tolerance on tooth flank.

Body shape

Design of cutter body shape is supported by selecting a shape and entering input parameters, which are located in the right expandable panel (Figure 6.18).

Body shape parameters are usually obtained from the database of pre-calculated standard bodies, by selecting body shape, type and bore size. Custom body could be constructed from the user input data. There are four types of body shape, that designer could attach to the cutter: disk, extended back body, hub and shank. Disk type tool design is shown in Figure 6.18(a), and shank is in Figure 6.18(b).

Tool face drawing

Missed teeth and solid spaces on the tool face could be defined by selecting *Tool face* tab in the central area of the main program screen (Figure 6.19). An option panel available on the right side contains parameters for teeth and spaces patterns, as well as allows defining custom tool face pattern.

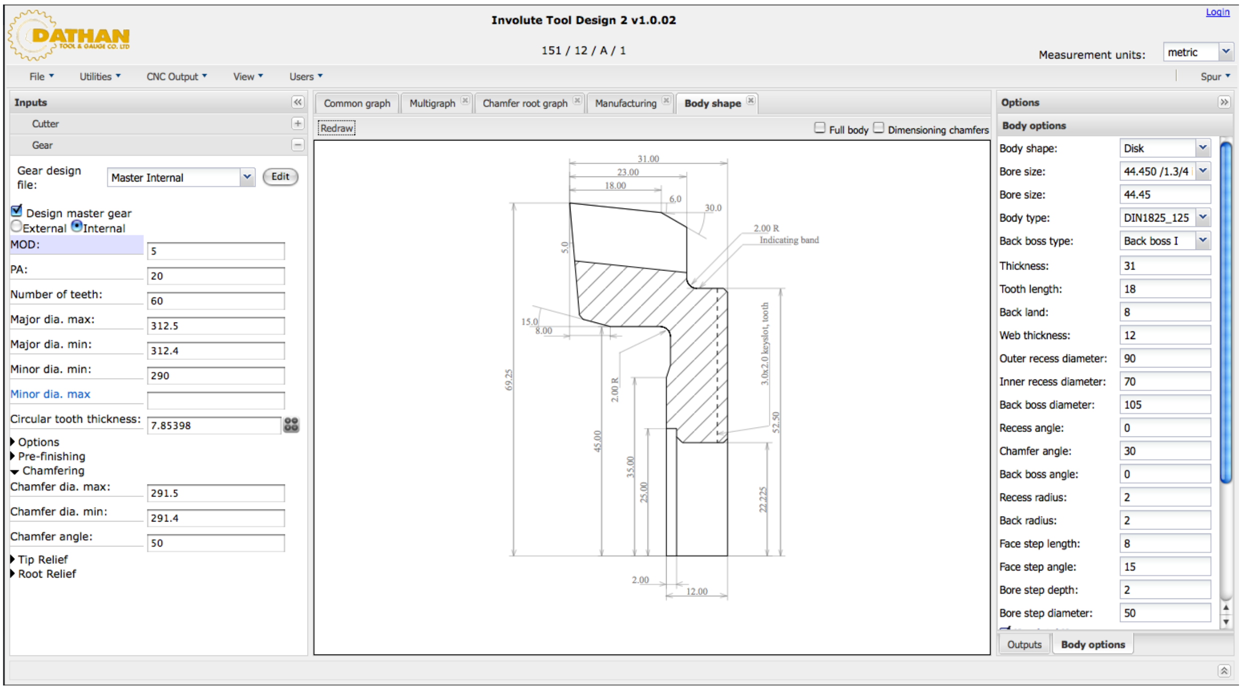
An option to display teeth order on the graph allows avoiding confusion while defining missed teeth.

Additional design information

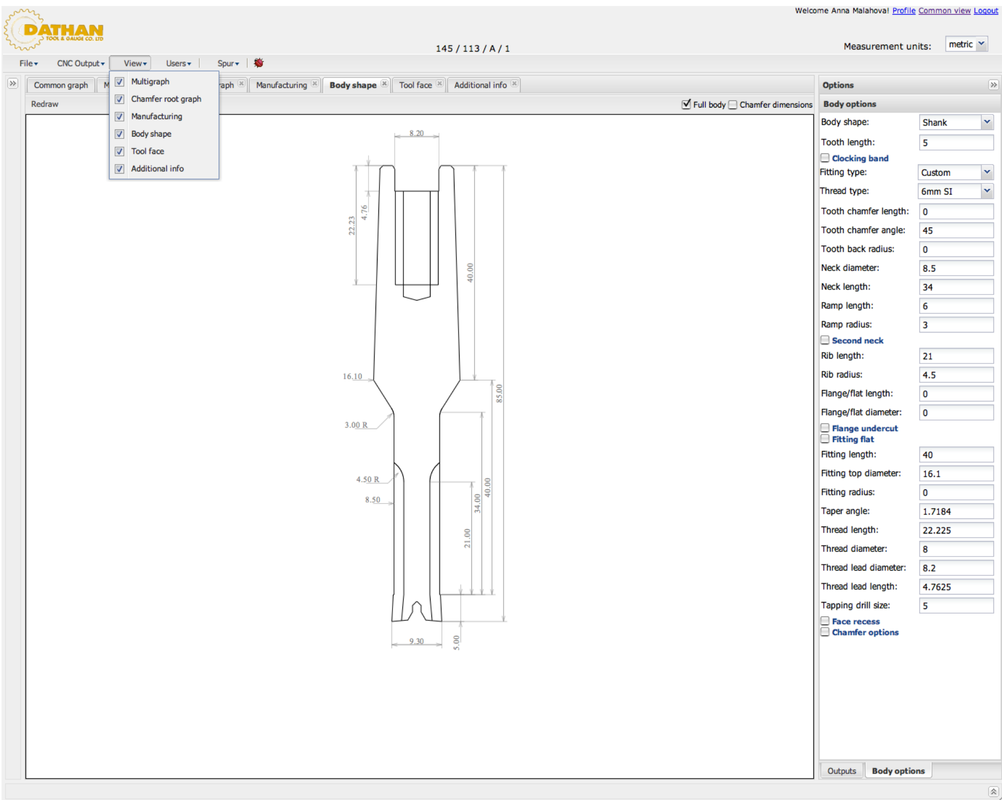
Inputs for additional design information, such as customer data and etch text, as well as PDF export options are located on a separate tab presented in Figure 6.20. Default etch text could be generated automatically from design data and then altered manually if needed. Etch usually contains information describing particular tool: module, pressure angle, number of teeth and design reference.

Additional design information tab allows input scale for exported PDF drawings, as well as define tool drawing PDF content. Multtooth former enables producing drawings and CNC code for formers with two or more teeth.

All drawings in the central tabs are automatically scaled to fit available space in the dedicated viewport. The left, right and bottom panels are resizable and could be minimised to leave more space for graphs.



(a) Disk type tool body shape



(b) Shank type tool body shape

Figure 6.18: Cutter body shape design. Body drawing is produced based on the inputs in the collapsible panel on the right side of the screen

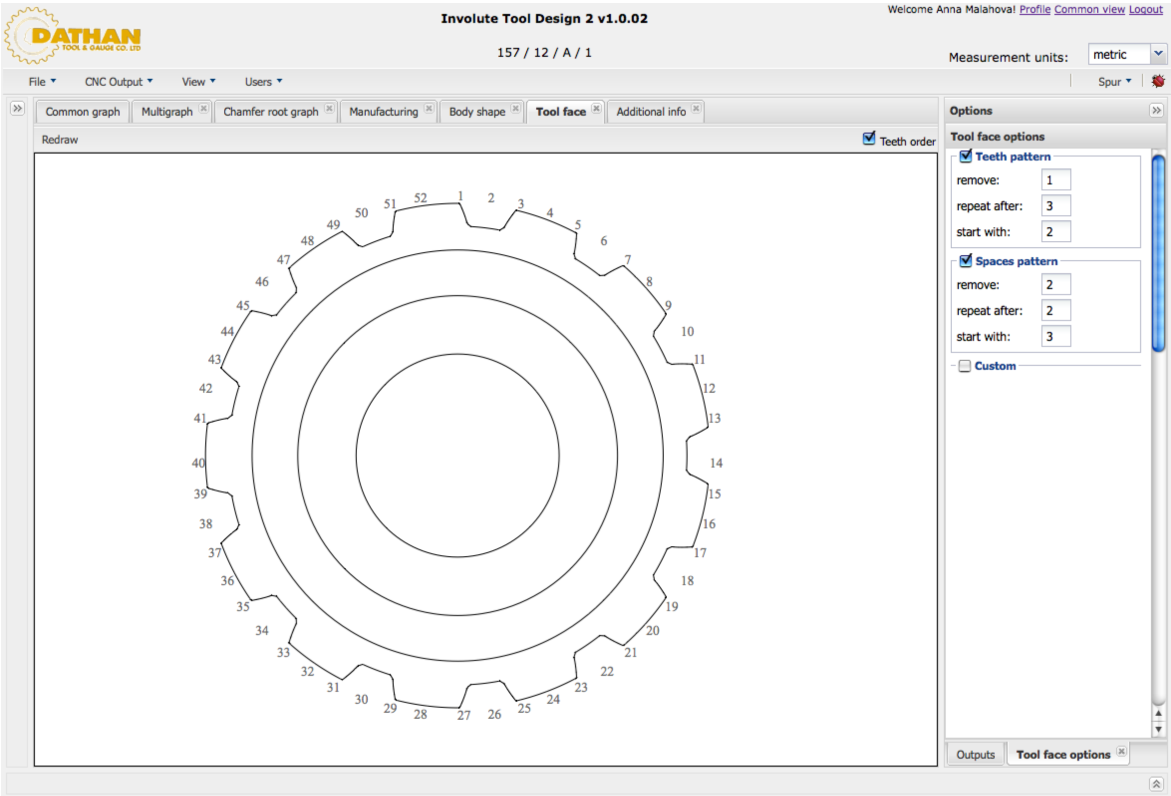


Figure 6.19: Tool face drawing. Inputs in the collapsible panel in the right side of the screen define which teeth or spaces to remove on the tool face. Cutter teeth are numbered clockwise in the drawing for the ease of reference in custom patterns

Dialogue windows

System dialogues are used to perform some operations. The most of dialogue windows in the systems are implemented as modal windows, i.e. they block user interaction with main application, when dialogue is active. Unlike other dialogues in the system debug panel is implemented as modeless dialogue window allowing the user to continue interaction with the main program, while enabling to select certain processing features.

Open design

A dialogue shown in Figure 6.21 will appear, when the user selects an option to open previously saved design from menu *File*.

The dialogue allows the user to review a list of previously saved designs, select one of them and load its data. The list of designs, stored in the database contains design reference, which consists of four parts: page number, book, revision and index; as well as design comment, its owner and date and time, when design was created and last modified. The list could be sorted by each of the design reference component: page, book, revision,

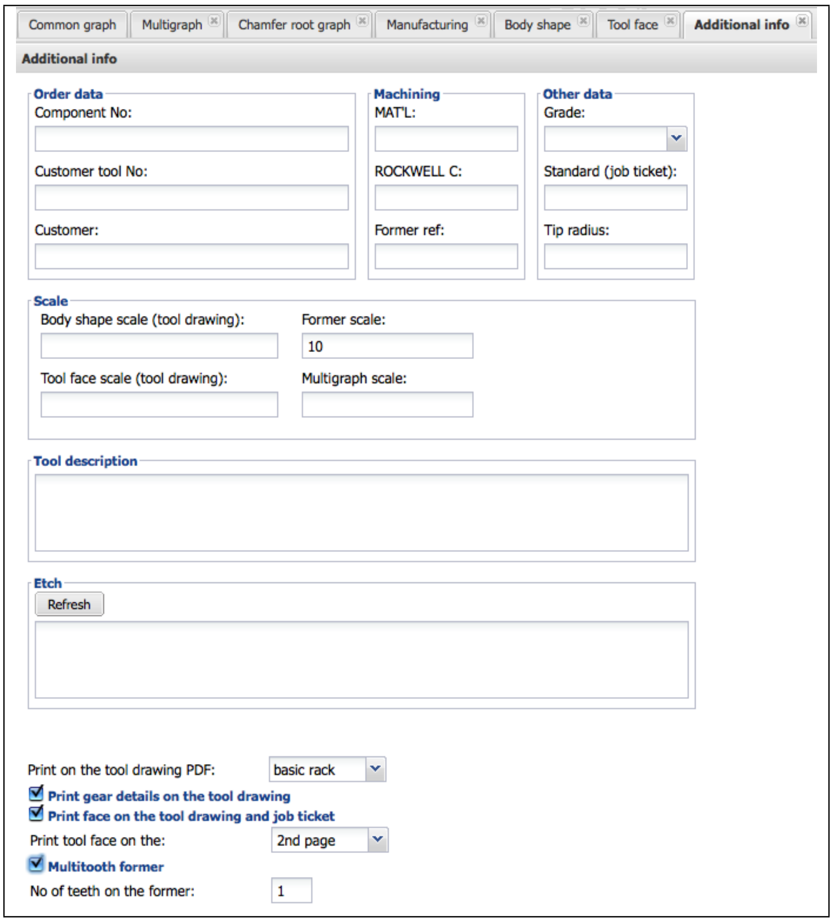


Figure 6.20: Additional design information

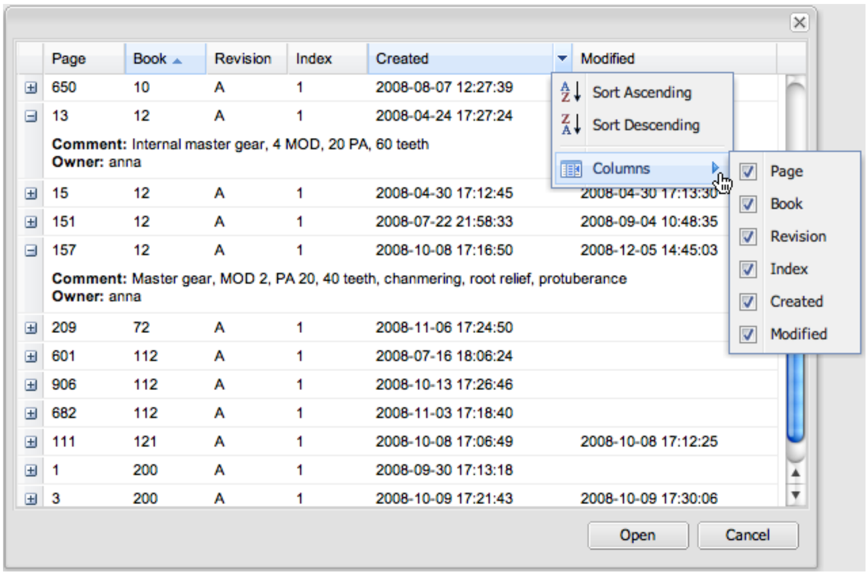


Figure 6.21: Open design dialogue window

number or by the date of design creation or modifications made. Information about design author and a comment are hidden by default and could be expanded if needed.

Save design

Design save dialogue (Figure 6.22) will be shown to the user when one of the options *Save*, *Save As* or *Save Copy As* is chosen from the menu *File*. This dialogue contains a list of previously saved designs, similar to the list of designs in *Open Design* dialogue. Double click on a design in the list will automatically offer to rewrite selected design data.

Dialogue has also five free input fields for entering design reference (page, book, revision, index) and comment. Unlike input fields, which define design reference, design comment input is optional. As combination of page, book, revision and index must be unique, reference check is done each time, when the user selects to save the design.

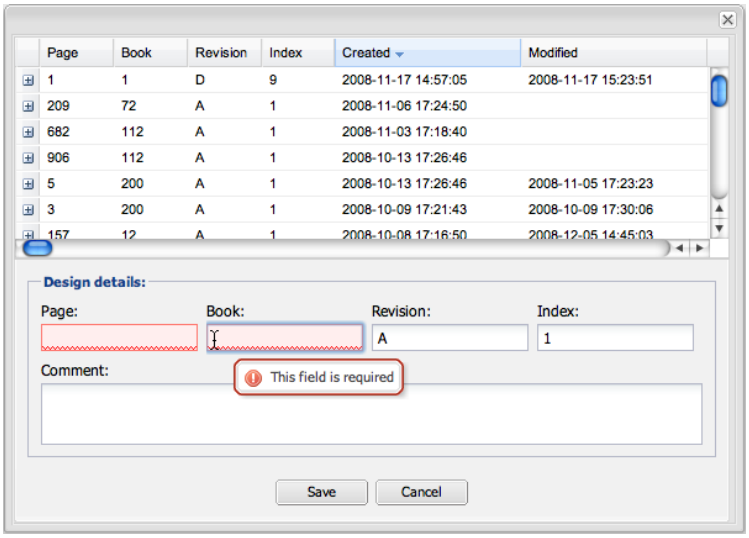


Figure 6.22: Save design dialogue window

Edit gear list

As a single cutter could be used to produce different gears, a design file can contain several gears. Adding gears to the design is performed using Edit gear list dialogue, which is presented in Figure 6.23. The dialogue window is shown to the user when a button *Edit* next to the gear file combo box is pressed.

Edit gear list dialogue contains a list of all gears related to the design and control elements, which allow adding new or removing existing gears from the list. Gear list is presented as a table with two columns: *Name* and *Comment*. A new record with automatically generated unique gear name is added in the gear table, when a button *Add* is pressed. User can edit the content of each cell in the table and all changes are marked

Figure 6.23: Edit gear list dialogue window

with red. Gear list edit dialogue offers a user two options: close dialogue and save all changes or close it and reject all changes.

Gear circular thickness calculator

Gear circular thickness calculator is implemented as dedicated dialogue (Figure 6.24). This module enable to calculate the circular thickness of a gear in three different ways depending on initially provided gear parameters.

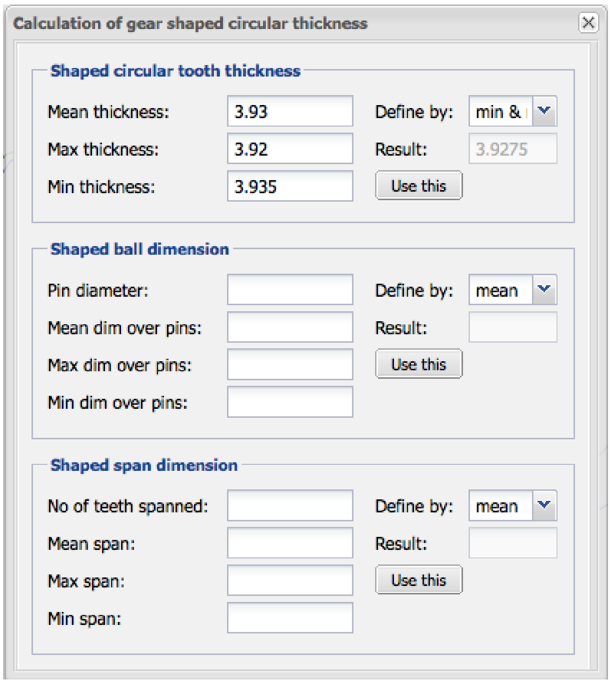


Figure 6.24: Gear circular thickness calculator

The user is expected to enter necessary parameters and choose to use calculated result for gear circular thickness input. There is an option to simply close the dialogue. Next time the module is open all inputs, which have been entered by the user in this dialogue, will be restored.

Edit manufacturing tool data

The data describing manufacturing tools listed in the right expandable panel when *Manufacturing* tab is active (Figure 6.17) could be altered or added in the edit manufacturing tool dialogue window (Figure 6.25).

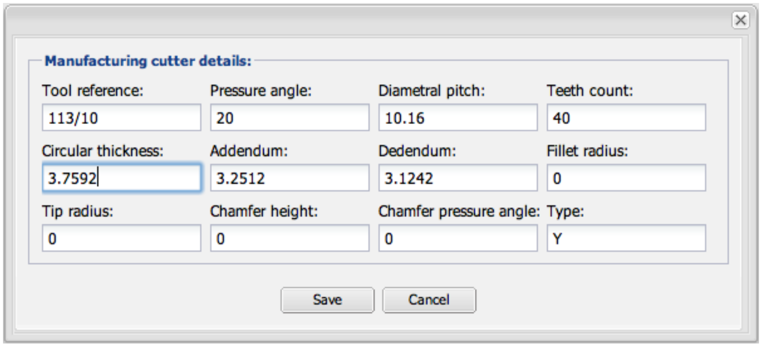


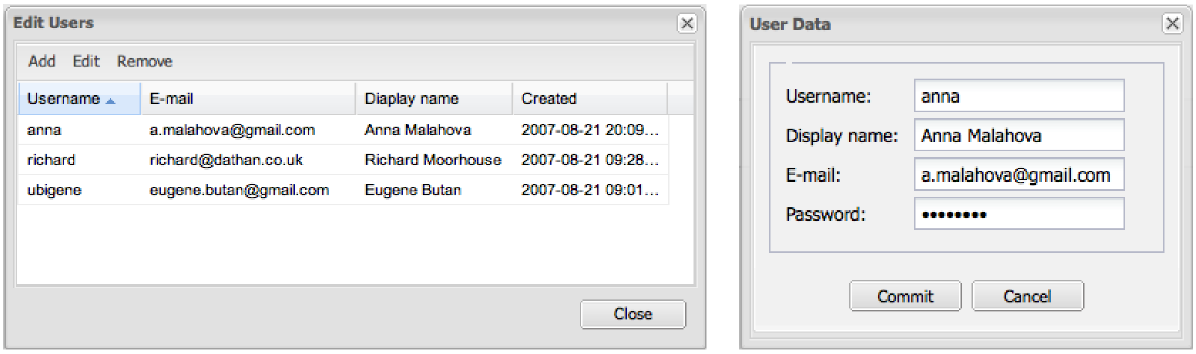
Figure 6.25: Dialogue to edit manufacturing tool data

Parameters, which define manufacturing tool, include tool reference, pressure angle, diametral pitch, teeth count, circular thickness, addendum, dedendum, fillet radius, tip radius, chamfer height, chamfer pressure angle and type. Data sufficiency and tool reference uniqueness check is performed when the user selects save the data to the database.

User account administration

Menu *Users* is available only for users with administrative permissions and has two options: *Users*, which enables user account data management, and *Groups*, which defines user membership in groups. *Edit Users* dialogue window is shown to the user, when option *Users* has been selected from the menu. The dialogue is presented in Figure 6.26(a) and contains a list of all users registered within the CAD/CAM system and options to add new user data, modify existing user account information or remove user account data from the system database. The list of system users is presented as a table with four columns: username, e-mail, display name and date, when the user account was created.

User data form shown in Figure 6.26(b) is used for adding new user account or modifying registered user data. Changes made in this form could be either committed or cancelled.

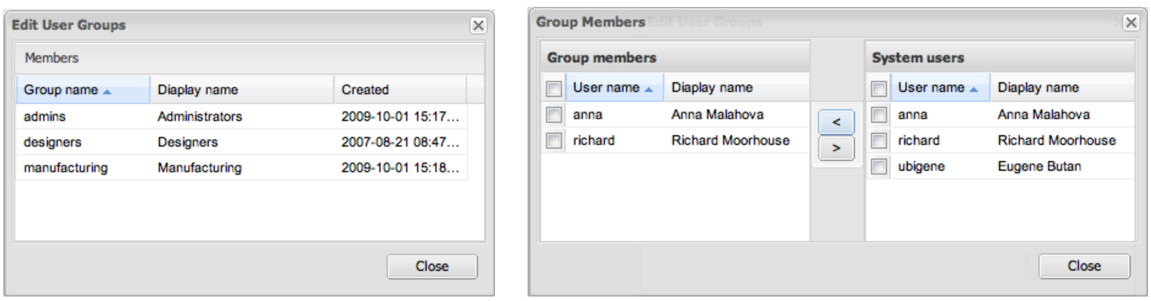


(a) A list of system users (b) Add/edit user data form

Figure 6.26: User account administration

User group membership management

Edit User Groups dialogue presented in Figure 6.27(a) window is shown to the user, when option *Groups* has been selected from menu *Users*. This window contains a list of user groups, which define user permissions to access different parts of the system and perform certain actions. The list of user groups is presented as a table with three columns: group nae, display name and date, when the group was created. User group display name could be modified by double clicking a corresponding cell and editing its content.



(a) A list of user groups (b) User group members

Figure 6.27: User group membership management

In order to define members of a group, the user selects the group from the list and presses button *Members*. This triggers *Group Members* dialogue window, displayed in Figure 6.27(b). The dialogue contains a list of selected group members, a list of all users, registered within the system, and controls, which allow adding and removing members from selected group.

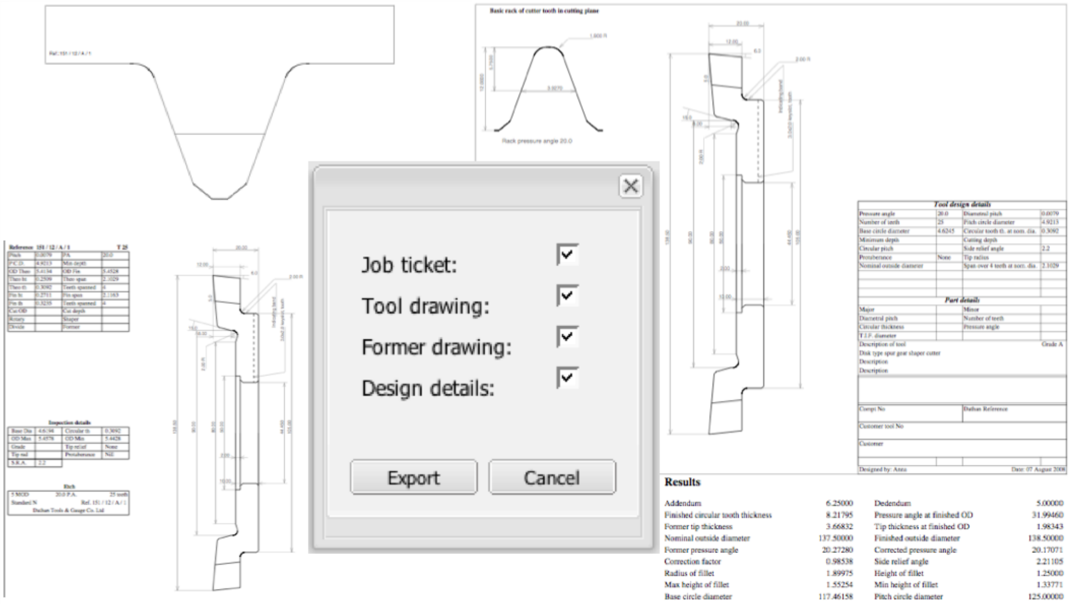


Figure 6.28: Exporting drawings in PDF

Export PDF

Menu *File* option *Export PDF* enables download four different types of automatically generated PDF documents:

- *Job ticket* contains data necessary to support manufacturing process of the gear cutting tool and includes technical drawing of tool body and tool face, if missed teeth or solid spaces pattern is defined for particular cutter.
- *Tool drawing* is used to present designed cutter specification to the customer and contains cutter and optionally gear details, basic rack or cutter, or both rack and cutter profiles, and body shape drawing. In the case of missed teeth of solid spaces on the tool, tool face drawing is included in the tool drawing as well. Designs with defined protuberance include protuberance diagram in the tool drawing automatically.
- *Former drawing* is used to create former, which will be used for producing designed cutter. The drawing contains former details and profile drawing.
- *Design details* contains initial design parameters and design outputs, describing particular design case. Master gear design details apart from design parameters include gear chamfer root diameter graph.

Dialogue window with corresponding options is presented in Figure 6.28.

Export and import from file

In order to enable design data transfer between two Web-based CAD/CAM systems running on different servers and using different design databases, the system has export to file and import from file options. This allows the user to save design data to a local file, which could be further imported to another system. Import from file dialogue is shown in Figure 6.29.

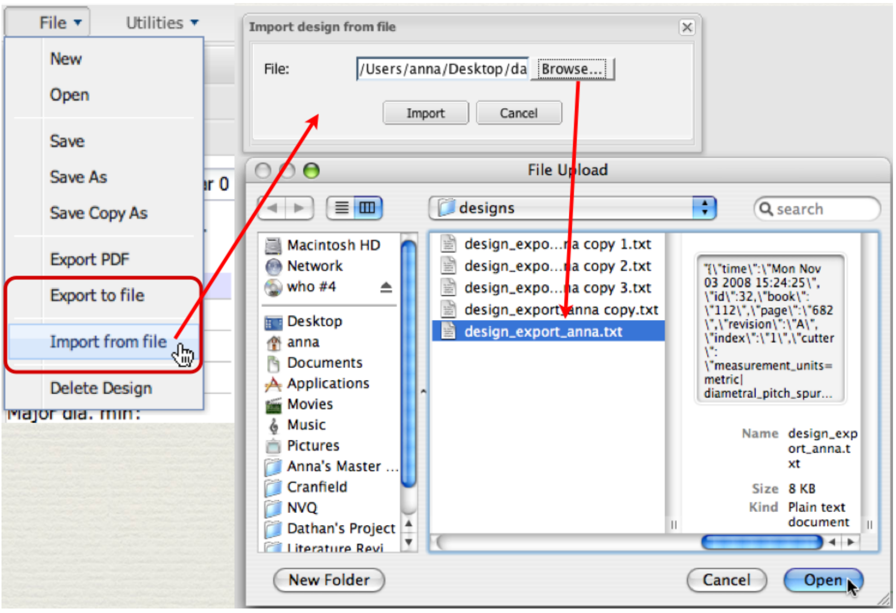


Figure 6.29: Design import from file

CAM

CAM portion in the system is presented as an option of automatic CNC code generation. Upon completion of gear shaper cutter design, the designer is offered to produce CNC code in one of two supported formats: Haas or Diaform (Figure 6.30). This operation automatically produces a CNC file with G-code in selected format and adds it to the manufacturing database.

6.9 Challenges and solutions

Applying the proposed methodology for the Web-based CAD/CAM system development case study encountered a number of challenges associated with upstream optimisation of

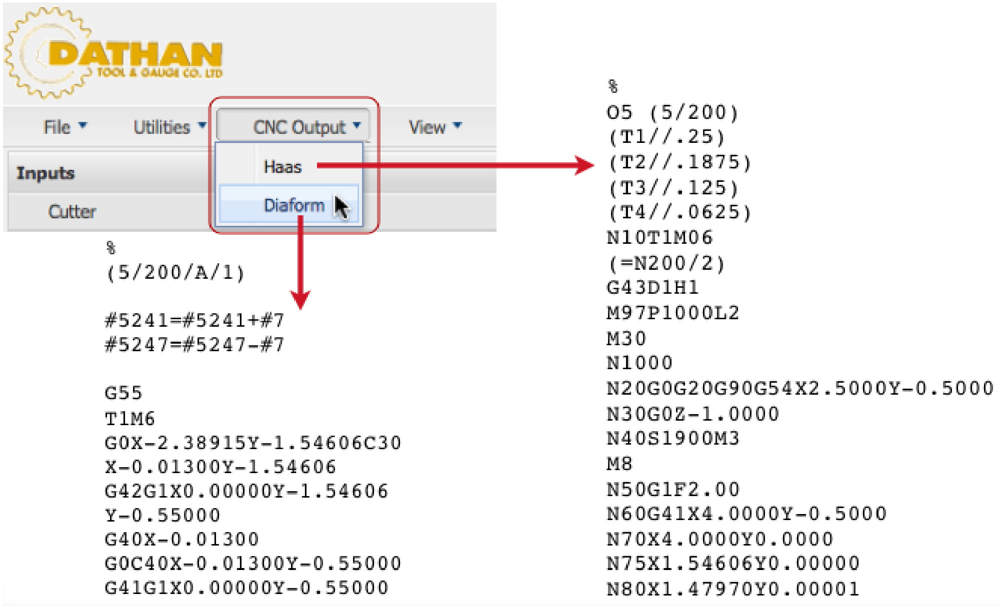


Figure 6.30: CNC code generation

the software architecture, exploratory development and process management. The ways of how those challenges were overcome are discussed in this section.

6.9.1 Evolution of the applied analytical model of gearing

To be able to effectively design gear shaper cutters, a designer should be able to analyse the gear tooth profile generated by the gear shaper cutter. The cutter and gear tooth profiles represent complex geometric shapes, each consisting of a number of different curve segments. The complex geometric profile of the gear tooth is specified by the generation process, translating complex geometry of the cutter tooth onto the gear. A dedicated CAD software would normally implement some sort of an analytical gearing model, which allows provisioning the necessary cutter tooth profile modifications and accounting for the changes in the cutting tool profile geometry throughout the life of the tool. A flexible design process, supporting various tooth profile modifications and cutter tooth profile optimisation for better performance throughout its life, would result in a complicated analytical model.

To deal with the increased complexity of analytical model for customised cutter tooth profiles the first prototype of the Web-based CAD/CAM system for gear shaper cutters introduced the full form technique, implying that the initial profile is considered as a whole regardless of the number and type of the profile modifications applied. Therefore, the tooth profiles are established as polylines (polygonal chains). On a downside, the full form generation technique produces conjugate profiles that may be crossed over themselves and therefore of a limited use to the gear designer. Therefore, the major precondition for the application of the full form generation technique is the ability to remove crossed over sections from the conjugate profile. For this purpose the algorithm developed by Shamos and Hoey was used.

The full form generation prototype established extremely flexible and robust gearing model. The tooth profiles produced using the first CAD/CAM system prototype appeared on average over the length of the profile very close to the real tooth shape. Thus, the prototype implementing the full form analytical gearing model has been approved for the further development of the CAD/CAM system.

Later the verification of the design results produced with the Web-based CAD/CAM system brought out the necessity to assess the accuracy of the tooth profile geometry. As it is very hard for a human to assess the overall accuracy of a geometrical shape, curve breakpoints serve as orienteers to guide the geometry verification process. The verification accuracy depends greatly on the precision of the curve breaks.

As it has been already noted, the full form generation approach uses polyline approximation of the composite tooth profile, assembled from the different curve segments. Therefore the precision of the calculated coordinates of the curve breakpoints and the performance of the CAD features depends on the number of straight line segments on the polyline. The number of points on the tooth profile would mostly affect trimming calculations and cusps removal from the profile, fidelity of involute graph and break points, as well as general responsiveness of the CAD/CAM system. In addition, the initial application of the Shamos-Hoey algorithm aimed to remove crossed over sections from tooth profile served well only for relatively simple profiles with limited number of intersections. An effective segment intersections detection algorithm was required to deal with complex profiles. In regard to this problem Shamos-Hoey algorithm has been replaced with Bentley-Ottmann algorithm implementation, chosen for its simplicity and low memory requirements. The improvement of the algorithm for finding line segments intersections provided a significant performance increase (by several orders of magnitude) on the native algorithm, but didn't solve the problem completely.

As mentioned before, the Web-based CAD/CAM system uses analytical model to obtain conjugated tooth profiles. Having a constructed rack profile is enough to calculate the associated cutter tooth profile, and having the latter it is possible to obtain the conjugated gear tooth profile. Herewith, the full form generation technique doesn't produce the conjugated tooth profile directly, but a set of points, corresponding to the shape generated in the blank, which can have crossed over sections near some of the curve breaks. Considering that the tooth profile is represented as a polyline, this means, that for finding a breakpoint on the tooth profile, an intersection of two straight line segments is calculated instead of finding the intersection of two curve segments. Reducing the length of the line segments on the rack profile by increasing the number of points on the polyline would result in more accurate cutter tooth profile and subsequently gear tooth profile, but the curve breakpoints would be still slightly out of place, as those never lie on the actual curves.

Moreover, regularly distributed points on the rack profile do not produce regularly distributed points on the cutter tooth profile, intensifying the irregularity of the line segments on the gear tooth profile. Therefore achieving acceptable precision of the curve breakpoints on the gear tooth profile appears a challenging and compute-intensive task, when applying the full form generation approach. It turns out that the only way to find the actual points of intersections between curve segments and thus ensure the required precision of curve breakpoints is to calculate those specifically.

Keeping in mind the stated problem the final version of the analytical gearing model has been altered in such a way, that allows to calculate all the curve breakpoints on the conjugate profile first and construct the rest of the tooth geometry with the number of points, that is enough for the smooth representation of the curves. The final gearing model implemented in the Web-based CAD/CAM system still utilises all the advantages provided by the full form generation technique and ensures good performance for the affected operations.

Summarising the experiences with the development of the applied analytical model of gearing for the Web-based CAD/CAM system for GSCs, it is important to note the role of the throwaway prototyping for the evaluating the system architecture. The prototype implementing the full form technique enabled to estimate the risks and benefits of this approach early in the development, thus minimising the amount of unnecessary rework, associated with the implementation of heavy, inflexible and complicated analytical gearing model.

Despite the analytical gearing prototype has been developed, it did not eliminate the associated uncertainty completely. So, in the beginning of the first case study the implementation of the curve breaks display and the precision of the breakpoints was not identified as critical. Also it was not clear at that point that the quality of the tooth profile is defined by the precision of the break points and not by how close the calculated tooth profile is to the real tooth shape in average over its length. The hot-fixes increasing the number of points on the polyline, representing the tooth profile, as well as the implementation of more effective algorithm for finding line segments intersections was not able to meet the new requirements for curve breaks precision and indicated that the problem requires a closer view. And there is no doubt, that the implementation of changes, associated with the new user requirements for curve breakpoints display and accuracy, was greatly facilitated by the development of well structured program code and use of abstractions.

The case with the applied analytical model of gearing illustrates the proof-of-concept prototyping and conducting exploratory development for reducing modelling uncertainty. The evolution of the analytical model throughout the case study development demonstrates the response to the changing user requirements, as well as the management of variations in calculations intensity by the development of a robust solution for analytical modelling of gearing.

6.9.2 Extensive computations, debugging and maintenance

It was already said, that the gear shaper cutter design process is highly parametrised and requires extensive calculations. A CAD system prototype has been implemented at the beginning of the case study development to specify the workflow of computations. All the design information is calculated based on the limited set of inputs, sometimes involving intermediate steps to obtain the missing parameters and the sought results. The implementation of multistep interrelated design calculations using regular conditional programming turned out as heavy, inflexible solution, poorly adapted to changes and extensions.

To address this issue and ensure that the application's knowledge base is dynamically extended with calculated design parameters, the system implements an inference engine based on forward chaining. "The engine uses elements of artificial intelligence and includes unification procedure of current state and conditions and result evaluation procedure" [20]. This implementation of design computations provides some advantages over the regular conditional programming, namely simplified formulae definition and subsequently better system debugging and maintenance.

This way the exploratory prototyping enabled the early identification of problems, associated with low flexibility and challenged debugging of the computational module, and thus allowed timely eliminating of the relevant architectural drawbacks.

6.9.3 User interface development

The user interface of the Web-based CAD/CAM system for GSCs implies the need to display a large number of user inputs and controls, visual representation of design data and numerical design results. The design of this kind of user interface is a challenging task and could lead to an overcomplicated, user unfriendly application with poor usability. The task of user interface design for the Web-based CAD/CAM system for GSCs is aggravated by the incomplete and ambiguous user requirements specification.

In this situation prototyping was used for effective user requirements elicitation and initial assessment of the implemented user interface design strategy. Based on the user feedback on the early prototypes of the Web-based CAD/CAM system the modular user interface has been found to be optimal from the perspectives of user-friendliness, usability and further scalability. The amount of the design information and controls can be adjusted by means of collapsible and expandable panels, thus enabling the effective management of the program screen space and avoiding unnecessary confusion of users by excessive elements displaying all at a time.

The early user interface prototyping also facilitated the upstreaming of graphical UI design decisions. The ability to modify the business logic, data flow and graphical representation individually is essential for a Web-based application with an intricate user interface as it simplifies the design significantly and results in more flexible and maintainable source code. Therefore MVC architecture has been implemented in the evolving prototype of the Web-based CAD/CAM system for GSCs.

6.9.4 Extending user requirements

It is well known, that the appetite comes with eating. So it is in the software development: user requirements tend to expand as the development proceeds. Partially this is governed by poor user awareness about current software development technologies and what is actually possible to implement; and with the user involvement in the development the perception gaps are smoothed over and the user facility for more features naturally increases.

Software projects with high level of user requirements uncertainty are highly apt to extending the requirements during the development process. There were numerous examples of extended user requirements during the development of the Web-based CAD/CAM system for GSCs, one of them being the requirement for the design of the second neck for the shank type cutter body shape, which was identified after the body shape design section has been already implemented. This change also affected the module for adding cutter body shape dimensions to the drawing, as the previous simple algorithm for body dimensions display in some cases would produce intersected dimensions when the second neck is defined. Thus extended user requirements for the design of customised shank type tools led to the development of a method for fancy display of cutter body shape dimensions.

For the development of the extended requirement for the customised body shape design as well as the improving the module for displaying dimensions the development team used the time reserve, that the case study plan envisaged for this kind of tasks.

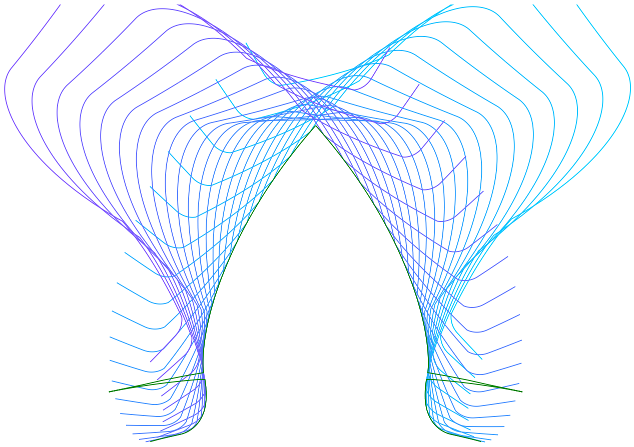
6.9.5 Tooth profile error problem

As it was already mentioned in Section 6.9.1, the analytical model for the gear shaper cutter design, implemented in the case study of the Web-based CAD/CAM system, employs full form generation technique to calculate conjugate profile. The model enables adding arbitrary profile modifications to the basic rack and producing the conjugate cutter and gear profiles without extending the analytical model for each modification.

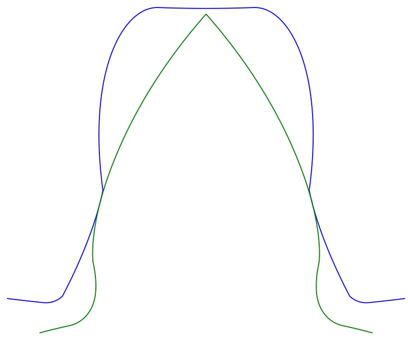
This is possible due to the approach, that considers the conjugated gear profile to be generated by the envelope of surfaces of the cutter. Apart from everything else, this idea allows for strict analytical determination of undercut conditions. A singularity of the envelope surface presents the condition for the undercut.

Considerable testing on real design cases detected conjugate profile errors produced in cases of undercut. An example of this kind of profile errors is shown in Figure 6.31. The research of this problem indicated a limitation of the implemented analytical model, related to the formula for the angle λ between the common tangent to the tooth profiles at the pitch point and the radius vector. The expression for finding the angle λ did not consider it's sign, which would change to negative on the profile segment, corresponding to the undercut.

The problem could have significant consequences in terms of the new analytical model applicability scope, and its priority was classified as major. Thus, a mini-research was conducted in the scheduled case study time to investigate the limitations of the analytical model, causing the tooth profile errors, and find an appropriate solution. The research, in particular, incorporated the analysis of profile errors and solving a mathematical task of finding the real value of the angle λ. The research resolved the undercut error problem and presented an improved analytical model of involute cylindrical gearing.



(a) Cutter roll and gear tooth profile with error on the undercut segment



(b) Conjugate profiles with fixed undercut error

Figure 6.31: Conjugate profiles with and without gear profile error on the undercut segment. The gear profile is shown in green and the cutter profile in blue. Different shades of blue correspond to various positions of the cutter tooth during the roll.

The successful implementation of this task required the development team to have the deep knowledge of the gearing geometry, strong analytical skills and expertise in solving applied mathematical tasks, as well as the ability to effectively communicate with other team members.

6.10 Impact of the development of Web-based CAD/CAM for GSCs

As the initial hypothesis underlying this thesis supposed that Web-based approach provides significant advantages for CAD/CAM software development over the desktop solutions, it is useful to review the effects from the development of the Web-based CAD/CAM system for gear shaper cutters.

Business process analysis shows that manufacturing volume before the development of the Web-based CAD/CAM system was limited to the number of designs, that the company was able to produce. Introduction of the new CAD/CAM system directly impacts the productivity and quality of tool designers' work, thus influencing the production volumes. The ability to produce flexible designs to meet customer requirements is affected by the implementation of new design methods and techniques, as well as the overall system performance and calculations precision. Implementation of modern data storage and retrieval technologies in the new Web-based CAD/CAM software system for accumulating and re-using knowledge about the designed products improves the expertise base of the company.

After all, utilising Web-based and particularly browser based solutions radically changes business processes, in particular by ensuring better accessibility, opportunities for user collaboration and customer involvement in product development process. The implementation of new CAD/CAM system can also impact other software used in the company, as it enables many integration opportunities.

6.11 Summary

The methodology for Web-based CAD/CAM software development proposed in Chapter 5 was used for the development of a Web-based CAD/CAM system for involute spur gear shaper cutters. In this case study functional and user requirements are not defined completely and accurately at the beginning of the development, but rather are gradually added and refined. This makes the process of software development hardly predictable and exploratory. The case study is also limited in time and people resources and overall fits well the main prerequisites for the applying the proposed methodology.

Several methods, which include user interviews with stakeholders, prototyping and obtaining feedback by demonstrations and conducting workshops, were used to elicit requirements for Web-based CAD/CAM system for GSCs include. To ensure adding, refining or changing of user requirements during the development, requirement elicitation activities were conducted continuously throughout the course of the case study.

Following the methodology proposed in Chapter 5, the development of this case study is organised in three sequential phases: initial phase, which incorporates software concept, risks assessment and initial requirements elicitation; software design optimisation and implementation of throwaway prototypes; and incremental development with numerous iterations. The last phase incorporates search for solutions when required. Section 6.4 provides a few examples of prioritising development tasks for each iteration and demonstrates on a few iterations, how the time reserve is managed inn accordance with the proposed methodology.

During the design of the Web-based CAD/CAM system for GSCs, the most critical decisions regarding the application interactivity, task distribution between the server and the client and multiuser interaction have been made considering the business requirements. By addressing the critical design decision points the most appropriate technologies and application infrastructure are investigated and application design is further optimised using throwaway prototypes. Then throwaway prototypes are developed for the most critical parts of the Web-based CAD/CAM system to evaluate the initial design and easily make changes if needed. In the Web-based CAD/CAM system for GSCs the most critical parts of the design include analytical gearing model, computational engine, user interface and visualisation and interactivity module.

A number of challenges were encountered during the application of the proposed methodology for the Web-based CAD/CAM system development case study. The challenges are associated with upstreaming the optimisation of the software architecture, exploratory development and process management. Solving these challenges in accordance with the proposed methodology confirmed its effectiveness for Web-based CAD/CAM software development.

During the development of Web-based CAD/CAM system case study the applied analytical model of gearing demonstrated the potential of exploratory development for reducing modelling uncertainty and the effectiveness of proof-of-concept throwaway prototyping for evaluating the Web-based CAD/CAM system architecture and estimating the risks and benefits associated with the implementation of a solution. Important architectural drawbacks were eliminated using exploratory prototyping, which enabled early identification of problems, associated with low flexibility and challenging debugging of the computational module. The evolution of the analytical model of gearing throughout the development of the first case study demonstrated the ability of the proposed methodology to respond to the changing user requirements, successfully deal with uncertainty and variations, and facilitate the development of robust solutions. Also, the early user interface prototyping facilitated the upstreaming of graphical UI design decisions, which resulted in flexible and maintainable source code and simple design, based on MVC architecture, providing the ability to modify the business logic, data flow and graphical representation individually.

Section 6.10 reviews the effects from the development of the Web-based CAD/CAM system for gear shaper cutters to prove the initial hypothesis underlying this research about advantages of Web-based approach for CAD/CAM software development over the desktop solutions. Implementation of the Web-based CAD/CAM system resulted in improved expertise base of the company, design flexibility, productivity and quality of tool designers work, thus increasing the production volumes. Moreover, utilising Web-based and particularly browser based solutions radically changes business processes, in particular by ensuring better accessibility, opportunities for user collaboration and customer involvement in product development process. The Web-based CAD/CAM system also enables many integration opportunities with other software used in the company.

The case study confirmed the applicability of the methodology and put to test many of its features in the circumstances of real development. Task prioritising and flexible project schedule enabled resolving of stakeholder availability challenges and ensuring good productivity of the development team by defining optimal level of stakeholder engagement using onsite or online collaboration capabilities. A number of challenges were encountered during the development of case studies, which provided possibilities to prove the effectiveness of the methodology for Web-based CAD/CAM system development.

Chapter 7

Web-based CNC code editor

This Chapter describes application of the proposed methodology for the development of a Web-based CNC code editor case study. The choice of this case study, like in the case with Web-based CAD/CAM system for GSCs, is based on the industrial focus of this research. The Chapter structure is organised based on the description of the methodology for Web-based CAD/CAM software, proposed in Chapter 5, and covers description of the case study, software concept and purpose, associated business processes and workflow, elicitation of user requirements, planning the development, resource allocation and collaboration, creating optimal design of the software, provides short description of the software architecture, implemented technologies and user interface. At the end of the Chapter, encountered challenges and implemented solutions are overviewed, impact of the development of Web-based CAD/CAM for GSCs is discussed and chapter summary is provided.

7.1 Case study description

Web-based CNC code editor is developed to enable making modifications to the CNC code, produced in the CAD/CAM system for GSC. The ability to modify automatically generated CNC programs is essential for reducing the impact of manufacturing process variations on the quality of the former, which is used for producing GSCs (Figure 7.1). CNC operators should be able to effectively eliminate shortcomings of the former elicited by inspection and make necessary corrections to the set of coordinates describing the path of the diamond tool.

A Web-based solution is preferred over the stand-alone implementation of the CNC code editor, because of better capabilities for integration with the other software used in

Figure 7.1: Generation of a helical cutter with a grinding wheel. The wheel shape corresponds to the former, which represents a corrected rack tooth profile [148].

the company, ability to store and re-use created CNC programs, simplicity of deployment and maintenance, cross-platform compatibility and low system requirements.

Although the Web-based CNC code editor is not a large-scale software application, the requirements identification is hindered by unclear and changing user vision of the final product. Therefore, the case study presumes the necessity of developing and verifying prototypes to explore all essential features of the software and ensure acceptable performance and correctness of the resulting CNC code. The new software development involves expertise in several areas and thus its success greatly depends on the clear and accurate requirements specification. Application of the proposed methodology could help to overcome many of these challenges.

7.1.1 Software concept and purpose

The purpose of the developing Web-based CNC editor is to improve quality of the produced tools, as well as reduce the time and effort spent on actual manufacturing. The new software is expected to improve the effectiveness of CNC operators' work, providing a convenient tool for making changes in CNC code, which is producing formers for gear

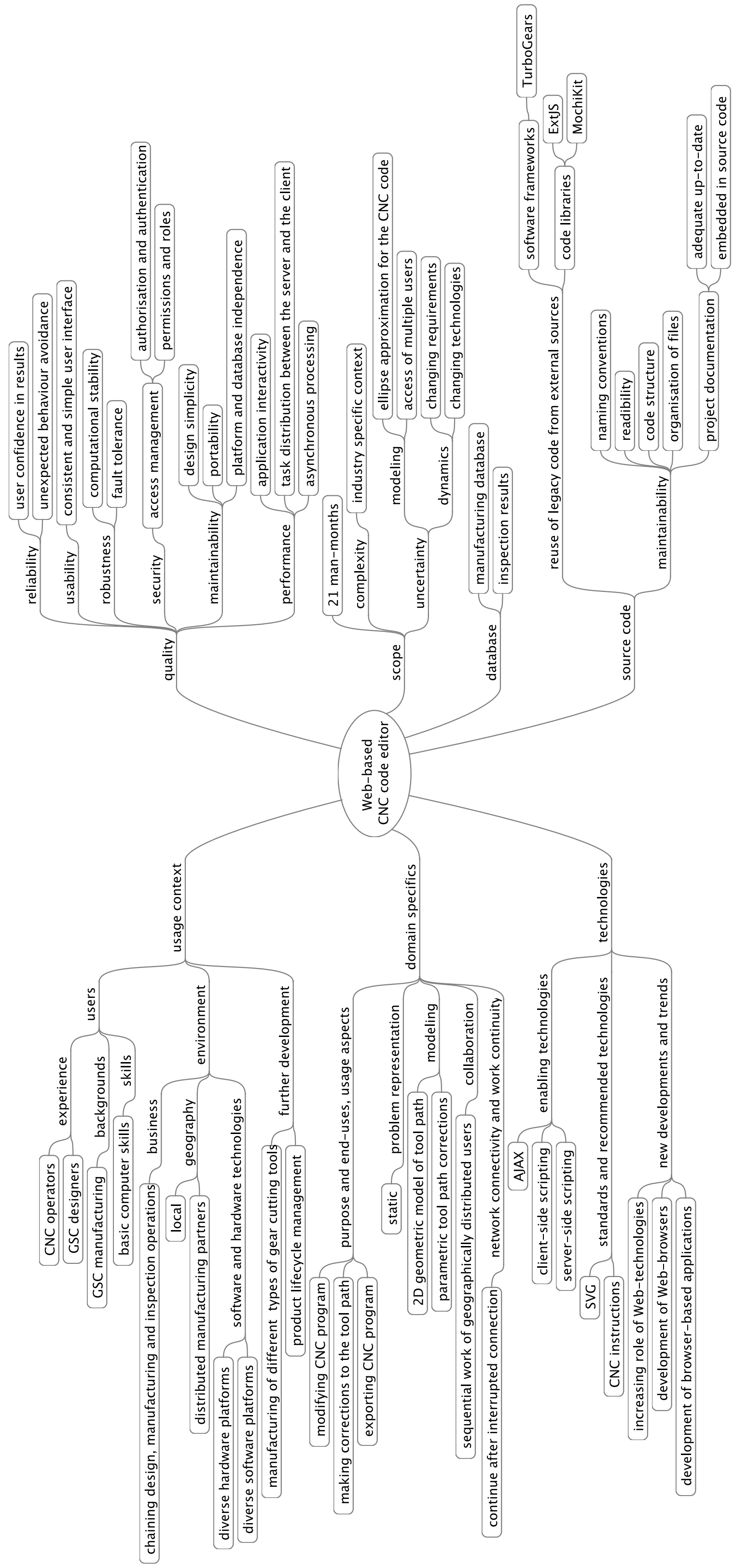


Figure 7.2: Mind-map containing different considerations related to the Web-based CNC code editor

shaper cutters. The editor should provide a visual model of the diamond tool path based on the set of 2D coordinates and enable manipulations with this model. Any changes to the model should be highlighted and the resulting CNC code altered accordingly. The Web-based CNC editor will be integrated with inspection and CAD software for GSC to enable automatic former geometry corrections with an option of manual override.

The specific objectives of the Web-based CNC code editor development include:

1. Facilitate the manufacturing and inspection process and reduce associated time and errors.
2. Develop an integrated Web-based CNC code editor to link with the other software used in the company.
3. Enable storing and re-using created CNC programs.
4. Ensure verification and plotting of the former geometry.
5. Ensure automatic profile corrections with an option of manual override.
6. Ensure Haas and Diaform CNC code support.
7. Ensure confidentiality and security when using the system.
8. Deliver well documented source code together with the finished software for the further development and maintenance.

Figure 7.2 sketches the initial vision of the developed software product taking into account different considerations related to the Web-based CNC code editor, such as its usage context, domain specific considerations, enabling technologies, source code and database aspects, as well as the scope of the case study and the desired quality of the final software.

7.1.2 Business processes and workflow

The process of CNC program editing includes input of CNC code and inspection data, if available, refining the geometry of the former and updating CNC instructions accordingly, which are then used to plot the tool path. The user can then edit the CNC program by either directly making modifications to the CNC instructions, or by manipulating the tool path model. In the first case the tool path model is updated considering the modifications made to the CNC code, and in the second case the CNC instructions are updated to reflect the changes in the tool path model. Finally, after the validation of the CNC program is successfully passed, the resulting CNC program is produced. The process model of the CNC code editing is presented in Figure 7.3.

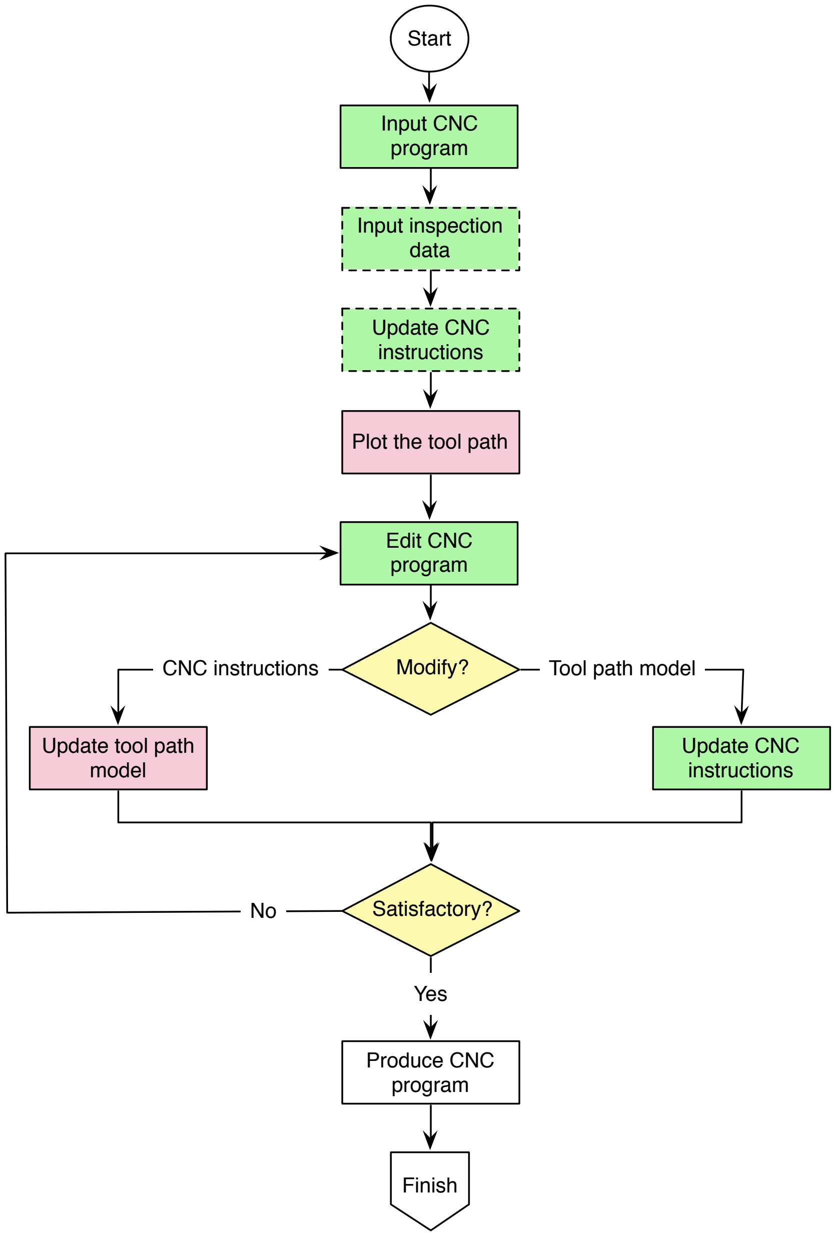


Figure 7.3: CNC code editing process

The CNC programs for producing gear shaper cutter formers are automatically generated uniform programs with identical machining process and cutting conditions. Former geometry is the most important part of the CNC program and commonly is the only subject to modifications. The modifications to the former geometry can be made automatically based on the inspection data or manually.

7.2 Eliciting user requirements

Similar to the Web-based CAD/CAM system development, main challenges associated with the CNC code editor user requirements elicitation are caused by unclear and changing user vision of the final software product and different expertise areas of system users and developers.

User requirements elicitation activities include prototyping and user interviews, which consider the following aspects of the new CNC editor:

- *CNC editor user interface*: implementation and look; CNC code editing mode; tool path plotting and manipulations; former geometry modifications; control elements layout; user roles and access management; usability; help.
- *Data input*: input fields layout and display; syntax highlighting; measurement units.
- *Saving and loading CNC program*: file saving and retrieval procedures; CNC program identifiers (file name, design reference); version control; data synchronisation; draft autosave option.
- *Integration with inspection*: inspection data retrieval, CNC program files version control.
- *Tool path plotting*: implementation of CNC instructions backplotting; required precision and scale; measurement units; validation.
- *Modifications to the former geometry*: manual CNC code modifications; tools for making direct modifications to the former geometry model; changes highlighting; reversing changes.
- *Parsing of CNC commands*: supported CNC syntax - Diaform, Haas.

Operators of CNC machines and GSC designers participate in user interviews and provide feedback on established system prototypes. User requirements elicitation activities are conducted continuously during the development of the case study to consider any refinements and changes.

7.3 Planning the development

Like in the first case study the development process of the Web-based CNC code editor incorporates three phases and its model is presented in Figure 7.4. In the initial phase software concept was developed, risks assessed and initial requirements elicited. During the second phase throwaway prototypes for user interface, tracking former geometry and former profile approximation with polyline were developed to find out the optimal software design. The last phase represents the incremental development of the Web-based CNC code editor, which undergoes 7 iterations, incorporating such activities as refining requirements, updating design documentation, assessing risks, prioritising tasks, short-term planning and adjusting time and effort estimation, refactoring, implementation, testing and fixing discovered errors, integration with the production system and obtaining feedback. During the incremental development search for solutions was used for the approximation of curved sections with circular arcs and mproving representation of former coordinates modifications for better readability.

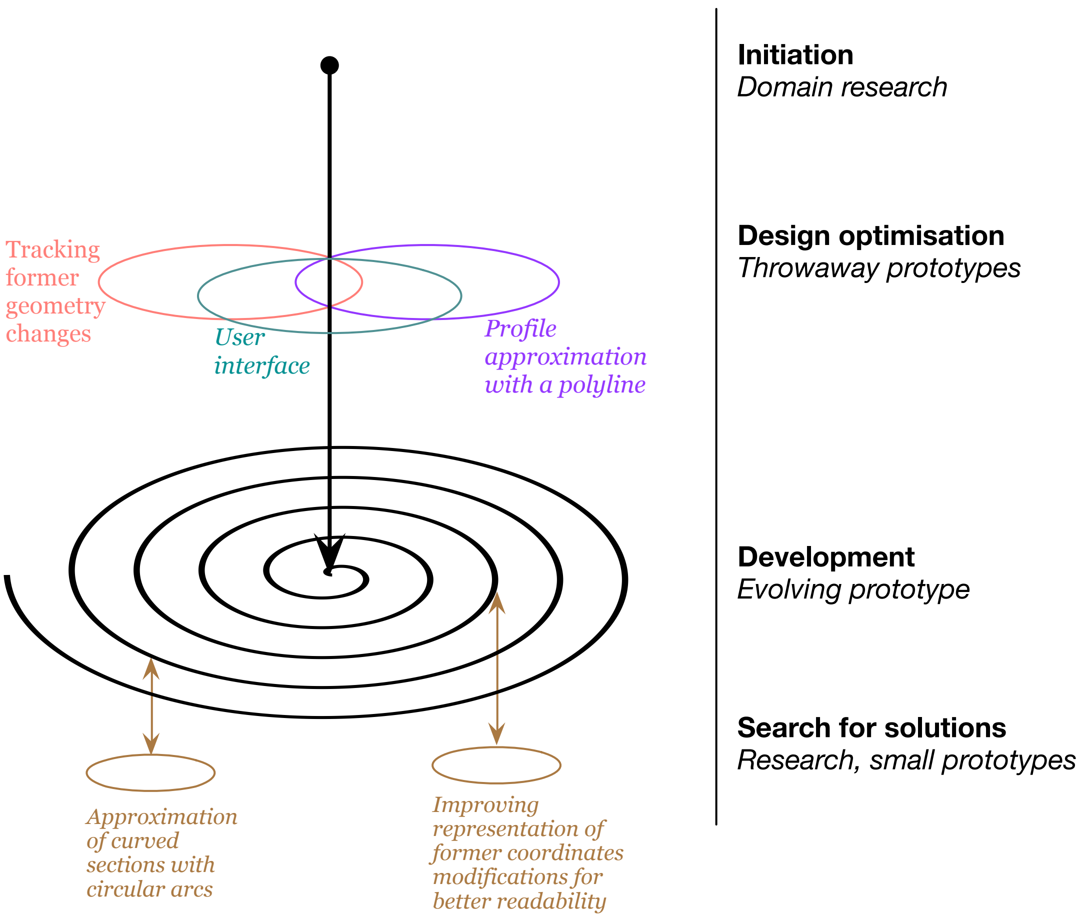


Figure 7.4: Process model for developing the Web-based CAD/CAM system for GSCs

In accordance with the proposed methodology, the case study development can be divided in three phases: initial phase, architectural decisions and incremental development. Figure 7.5 presents the work breakdown for the development of the Web-based

CNC code editor. The software development actions incorporate software concept, risks assessment, initial requirements elicitation, architectural decisions implemented using throwaway prototyping and incremental development, divided in 7 iterations.

#	Title	# Successors	# Predecessors
0	**Web-based CNC code editor**		
1	Project initiation	2	
2	**Software concept**	**6**	**1**
6	**Risks assessment**	**10**	**2**
10	**Initial requirements elicitation**	**13**	**6**
13	**Architectural decisions**	**27**	**10**
27	High-level structured design documentation	28	13
28	**Development planning**	**33**	**27**
33	**Incremental development**	**101**	**28**
34	**Iteration 1 – Back end**	**43**	
43	**Iteration 2 – Front end**	**52**	**34**
52	**Iteration 3 – CNC code parser**	**61**	**43**
61	**Iteration 4 – Tool path backplotting**	**71**	**52**
71	**Iteration 5 – Integration with inspection**	**81**	**61**
81	**Iteration 6 – Geometry modifications and changes synchronisation**	**91**	**71**
91	**Iteration 7 – User access management**		**81**
101	**Documentation**	**105**	**33**
105	**Testing**	**108**	**101**
108	**Roll Out**	**113**	**105**
113	Launch finished		108

Figure 7.5: Work breakdown for the Web-based CNC code editor development

With the help of expert judgement the effort required for the development of the Web-based CNC code editor has been estimated to be about 21 man-months, or case study duration equal to 7 months considering the development team of 3 people.

Similar to the Web-based CAD/CAM system for GSCs development, the reserve threshold for the Web-based CNC code editor is assumed $m = 30\%$, and the lag δ is set to 3% to avoid excessive fluctuations in the speed of reserve utilisation.

A few iterations are shown in Table 7.1 to illustrate the task prioritisation during the development of the case study. Tasks included in the core category are planned activities, absolutely required for the successful implementation of the case study. Major tasks are not critical for the success of the case study, but still associated with important

Table 7.1: Prioritising development tasks for the Web-based CNC code editor

Iteration	**Core tasks**	**Major tasks**	**Optional tasks**
1. Back end	Define data structures and data types, physical data organisation, data processing programs, build connections to the front end		
2. Front end	CNC code editing mode; input fields; main menu; file save and retrieval dialogues.	Tabular representation of the former profile coordinates for better readability.	Syntax highlighting for CNC code editing mode.
3. CNC code parser	Parsing CNC code instructions to vector geometry objects and vice versa; Diaform and Haas syntax support.	CNC code validation.	
4. Tool path plotting	Plotting of CNC instructions; control elements for selecting and deselecting points on the former profile; controls for manipulating the tool path model and its parts.		Diamond tool movement simulation.
5. Integration with inspection	Inspection controls; inspection data retrieval and former geometry corrections.		CNC program files version control.
6. Geometry modifications and changes synchronisation	Control elements for making direct modifications to the former geometry model; manual CNC code modifications.	Highlighting changes.	Reversing changes.

requirements. Optional tasks reflect additional improvements, that have least priority for the case study.

Considering the estimated time required for the Web-based CNC code editor development equal to 7 months for a team of 3 people and assuming the average number of working days per month equal to 22, the planned case study duration is 154 days and reserved time is about 46 days. The management of this time reserve is demonstrated on the example of six iterations, prioritised in Table 7.1 and further referred Table 7.2. During the first couple of iterations, when the back and front end of the Web-based CNC code editor were developed, the actual effort spent has been close to the given estimates and thus the relative part of the reserved time in the case study has been increasing until it exceeded the threshold tolerance of 10% in the beginning of the third phase. Therefore the implementation of optional tasks was enabled for the development of CNC code parser. The development of the third iteration took longer, than it was expected,

Table 7.2: Time reserve management for the development of the Web-based CNC code editor

Iteration	**Planned and reserved time for the rest off the case study,** *days*	**Current reserve** c**, %**	**Scheduled tasks**
1. Back end	Planned time = 154; Reserved time = 46	30	Core + Major
2. Front end	Planned time = 147; Reserved time = 46	31	Core + Major
3. CNC code parser	Planned time = 130; Reserved time = 44	34	Core + Major + Optional
4. Tool path plotting	Planned time = 110; Reserved time = 32	29	Core + Major
5. Integration with inspection	Planned time = 95; Reserved time = 26	26	Core
6. Geometry modifications and changes synchronisation	Planned time = 74; Reserved time = 25	34	Core + Major

sequently limiting the tasks scheduled for the next iteration to core and major activities only. In turn, the development of the tool path plotting also took longer, than it was planned for this iteration, subsequently limiting the development of the next iteration with core tasks only. These measures eventually led to the increased relative time reserve in the beginning of the sixth iteration, extending the scheduled activities with tasks, that are categorised as major.

This way the usage of the time reserve is being continuously monitored and adjusted accordingly to the current situation, ensuring the delivery of the most important and critical features on time. The use case of the Web-based CNC code editor development provides a real-life application of the mechanism for the time reserve management, described in the section 5.3.

7.4 Resource allocation and collaboration

Like in the first case study, the development team is created specifically for the Web-based CNC code editor development. The team involves people with expertise in software development, manufacturing of gear cutting tools and CNC. Problems that raised during the development, were solved by the development team, thus external experts help was not necessary.

Table 7.3: Engagement of different stakeholders in Web-based CNC code editor

Involvement	**Onsite collaboration**	**Online collaboration or teleconference**
Day-to-day development	Developers and experts in CNC and manufacturing, involved in the development. Participation of CAD/CAM experts as required	CAD/CAM experts
Weekly progress review	Developers and experts in CNC and manufacturing, involved in the development	CAD/CAM experts, executive manager and facilitator
Monthly progress review		Key stakeholders
Quarterly meeting	Key stakeholders	

Table 7.3 shows the intensity of engagement of different stakeholders in this case study development, which depends on their commitment and influence. Onsite and online collaboration capabilities are utilised to ensure optimal level of stakeholder engagement, which combined with flexible project schedule addresses stakeholder availability challenges and ensures good productivity of the development team.

7.5 Creating optimal design of the software

Based on the business requirements design decisions about the application interactivity, task distribution between the server and the client and multiuser interaction have been made. Interactive capabilities of the Web-based CAD/CAM system are enabled by handling application events with scripting languages, using synchronous and asynchronous requests for effective management of data inputs and outputs, using vector graphics and scripts for visualisation and manipulations with graphics, as well as optimising system responsiveness to fit within the performance and latency constraints and ensure desktop-like user experience. Interactivity related design decisions for the Web-based CNC code editor are similar to those for the Web-based CAD/CAM system for GSCs presented in Figure 6.5.

Collaboration capabilities of the Web-based CNC code editor are defined through sharing resources and knowledge and interaction in different time/same place and different time/different place, like it is defined for the Web-based CAD/CAM system for GSCs in Figure 6.6.

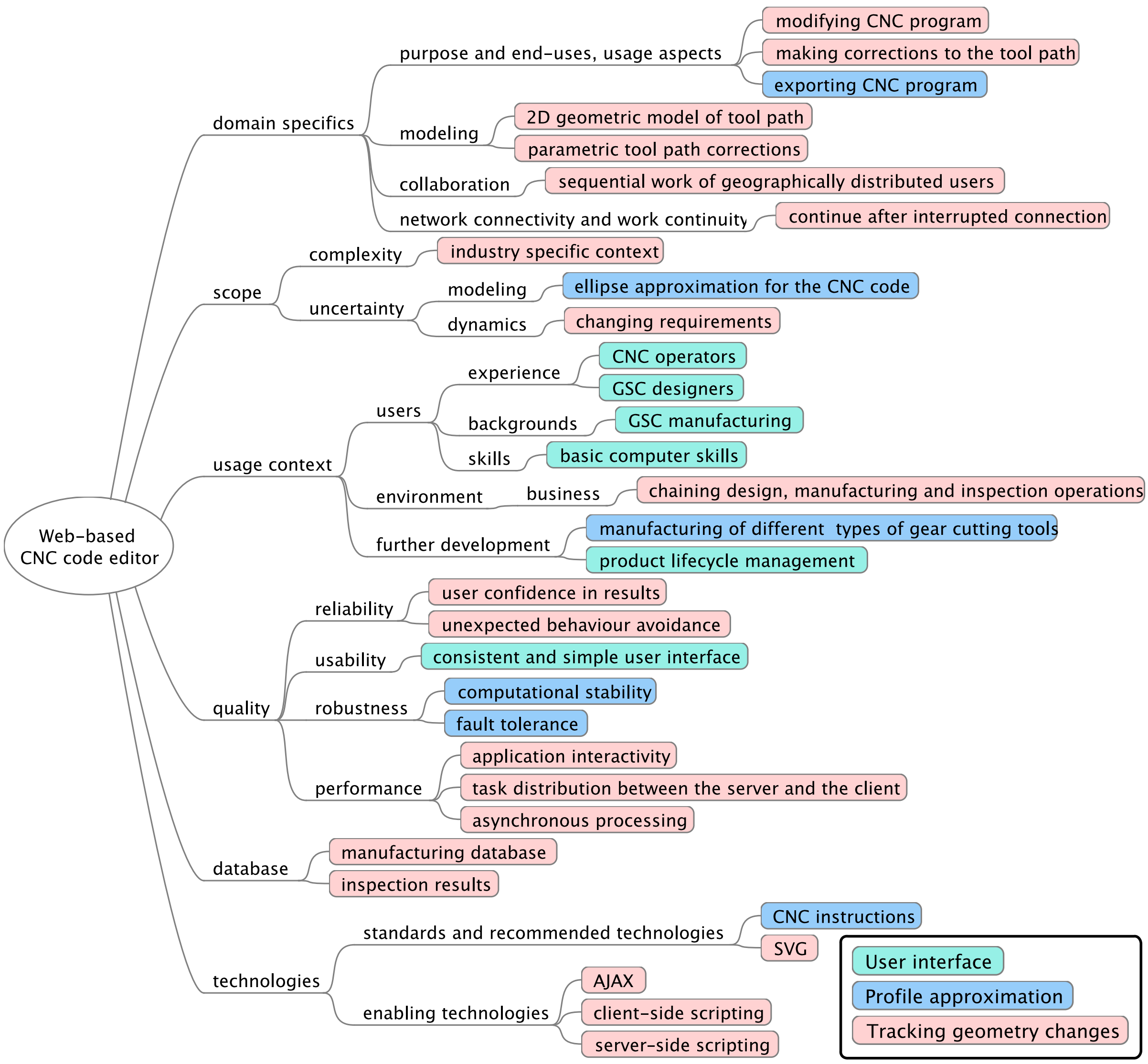


Figure 7.6: Addressing design considerations for the Web-based CNC code editor using prototypes for user interface, profile approximation procedure and module for tracking geometry changes

The distribution of tasks between the client and the server impact the performance of a Web-based system and should be addressed when creating optimal application design. Table 7.4 lists major application tasks designed to be executed on the server and on the client in the Web-based CNC code editor. For better application responsiveness operations directly related with interactions between the user and the system are executed on the client side, while data operations and computational tasks are performed on the server side of the Web-based CNC code editor.

Making major design decisions about application interactivity, task distribution between the server and the client and multiuser interaction helps to create initial applica-

Table 7.4: Task distribution between the server and the client in the Web-based CNC code editor

Server	**Client**
Saving and retrieving data;	Tool path plotting and manipulations;
CNC instructions backplotting;	Data input;
Synchronising modifications and changes;	Input data validation;

tion infrastructure and select appropriate technologies, which are the subject for further optimisation and approval with throwaway prototypes. To evaluate the initial design decisions of the Web-based CNC code editor were created throwaway prototypes of user interface, former profile approximation procedure and module for tracking former geometry changes. Figure 7.4 shows how the development of these prototypes is in line with the case study development process model. The developed throwaway prototypes cover some of the software product considerations mind-mapped in Figure 7.2. Those are related to the domain specific considerations, project scope, application usage context, software quality, database and implemented technologies. Figure 7.6 shows the coverage by the corresponding prototypes using colours.

Basic business rules and informational aspects of the subject area, associated with the development of the Web-based CNC code editor for gear shaper cutters are presented using ERM, which is shown in Figure 7.7. Information about CNC programs is stored in the database, which could be accessed by a worker from manufacturing department. The worker operates a CNC machine and uses stored CNC program to produce the former, which is later used to dress the grinding wheel. Finally the grinding wheel is used to grind the gear shaper cutter.

The produced cutter further is inspected and if necessary the quality of the cutter could be improved by refining the geometry of the former and correcting the initial CNC program.

The use cases of the Web-based CNC code editor are encompassed by three roles:

- *Designer*, who can access the Web-based CNC code editor and CNC programs database;
- *Administrator*, that is responsible for user permission management;
- *CNC operator*, who can access the Web-based CNC code editor and load stored CNC programs, modify them, apply inspection results and save the changes to the database.

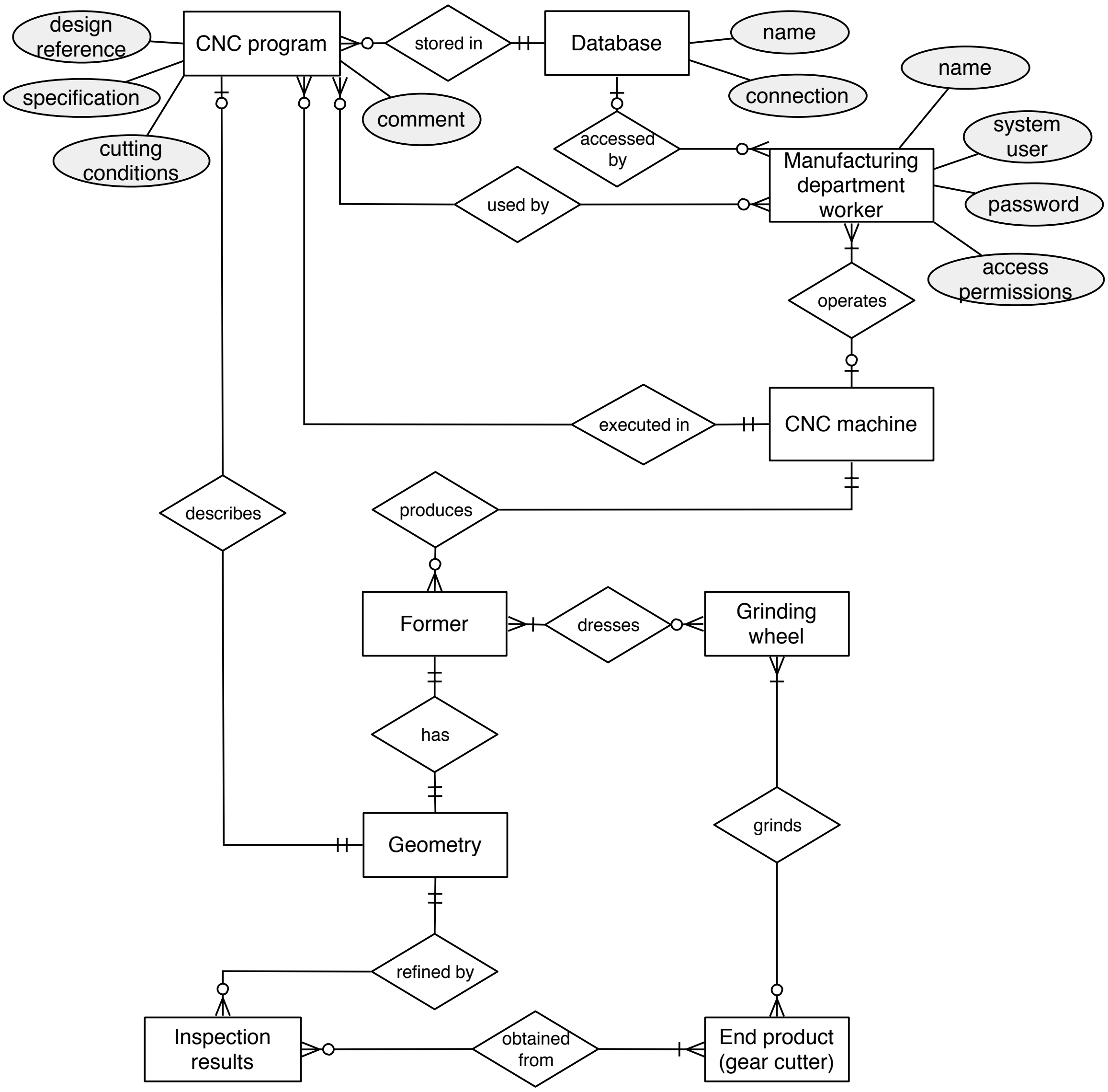


Figure 7.7: Entity-relationship diagram of Web-based CNC code editor for gear shaper cutters

The latter two roles extend the first one with specific interactions, as shown in Figure 7.8.

Similar to the Web-based CAD/CAM system for GSCs, handling of all the components of the Web-based CNC code editor and page reconstruction after any changes is achieved using MVC pattern. The concept of business logic separated from content and presentation enables independent development, testing and maintenance of each of these parts.

2D visualisation of the tool path is implemented using SVG, which allows to avoid using supplementary programs for displaying drawings within the Web-browser. Interactive

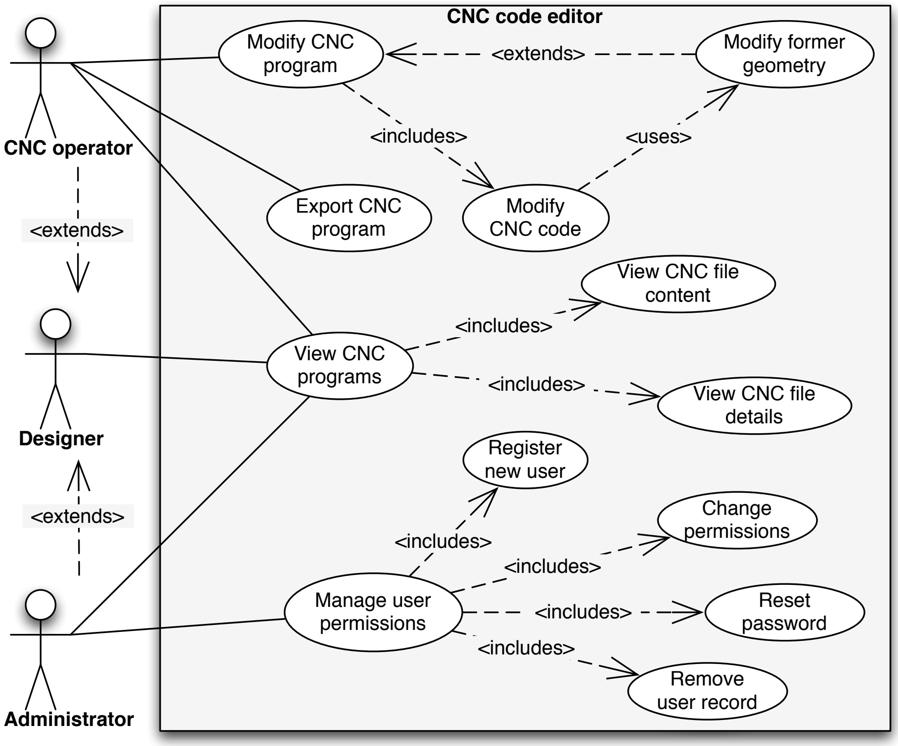


Figure 7.8: Use case diagram of Web-based CNC code editor for gear shaper cutters

manipulations with the visual model of the tool path are enabled by AJAX technology utilisation. Asynchronous data communications between the client and the server ensure independent display and behaviour of the Web page, thus imitating the desktop-like applications user experience.

7.6 Software architecture and implemented technologies

As the Web-based CNC editor has a number of key requirements common to the Web-based CAD/CAM system for gear shaper cutters, described in the first case study, and is intended to be used in conjunction with the CAD/CAM system to assist the production chain of the same GSC manufacturing company, the strategic design decisions for both applications has many similarities.

The architecture of the Web-based CNC code editor for gear shaper cutters is presented in Figure 7.9. Client requests **A** are dispatched to the server-side controller **B**, which collects all the data for the response and passes it to the template engines **C** to be turned into the representation that the user will see. Intermediate results **G** - graphical model of the tool path, which is implemented using the SVG format, and updated CNC code instructions - are obtained from the user inputs **F** by means of client-side **I** and server-side **B** controllers. The access to the inspection results **D** and the manufacturing database **E** is ensured using the server-side controller **B**.

Interactivity controller **J** enables real-time information updates when the user is performing manipulations with the graphical model of the tool path or he is making changes in the coordinates describing former geometry.

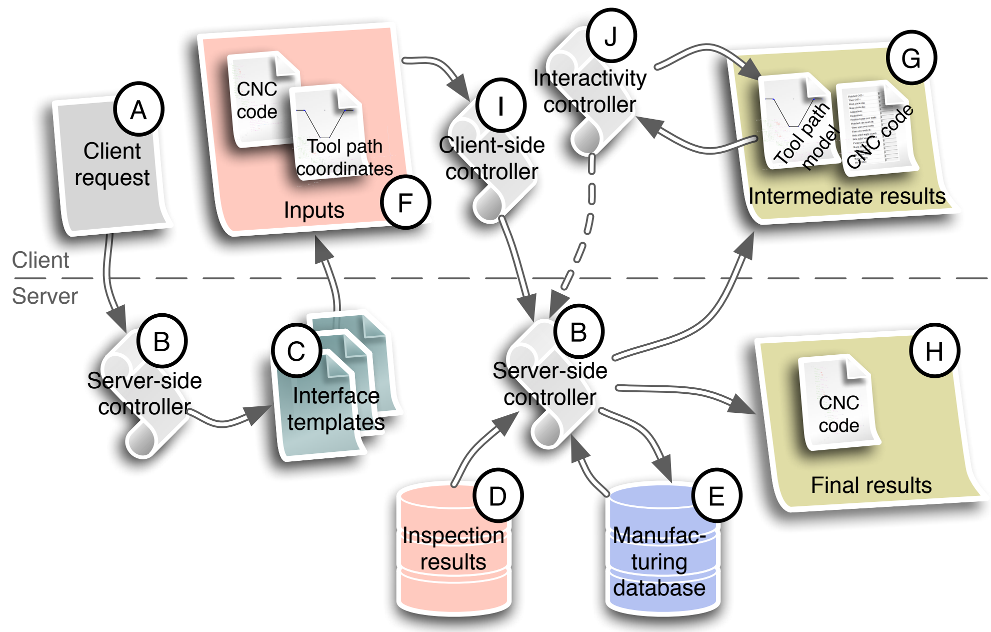


Figure 7.9: Architecture of the Web-based CNC code editor for gear shaper cutters

The Web-based CNC code editor for its interactive capabilities implements MochiKit and ExtJS JavaScript libraries, and utilises AJAX technology for data interchange between the client and the server.

The database is controlled by means of PostrgreSQL database management system, providing the required safety and performance and available for many platforms including Linux, FreeBSD, Solaris, Microsoft Windows and Mac OS X.

The server portion of the Web-based CNC code editor is implemented using Python programming language and TurboGears framework, which facilitates processing of Web-queries and ensures authentication services and run-time environment. Python in conjunction with TurboGears increases developer's productivity and lowers maintenance costs and is well suited for the development of software in a limited time scale. The technology choice also establishes consistency with the previously developed Web-based CAD/CAM system for gear shaper cutters.

Table 7.5: Deployment requirements

	Server	**Client**
Operational environment	• Microsoft Windows starting with Windows 2000, Mac OS X 10.3 or later, Linux starting with core version 2.4.12; • PostgreSQL 8.0 or later; • Python 2.4 (or later), TurboGears 1.1. • Backup software and hardware.	• Operating system able to run Safari 3.0 (or later), FireFox 2.0 (or later) or Google Chrome Web browser; • Safari 3.0 (or later), FireFox 2.0 (or later) or Google Chrome Web browser.
Hardware	• IA-32 or AMD64 architecture (compatible with architectures MIPS, POWER, PowerPC and SPARC); • CPU 1 GHz (3 GHz dual core system recommended); • RAM 1 GB (3 GB recommended); • RAID 1 300 GB; • Gigabit Ethernet.	• Any computer and operating system able to run Safari 3.0 (or later), FireFox 2.0 (or later) or Google Chrome web browser.
Communication requirements	• Internet connection starting with 1 Mbit (data rate might be higher for increased number of Web-based CNC code editor users).	

The Web-based CNC editor is compatible with different program environments and platforms; it can be installed on the server running Microsoft Windows, GNU/Linux or OS X, and accessible from anywhere through the Internet using a modern Web browser. The editor does not require any additional components installation on the client side for enabling full functionality.

As in the case with the Web-based CAD/CAM system for gear shaper cutters, the development of the CNC editor is realised on OS X 10.4 - 10.7 platform, using JEdit and Eclipse IDE and such tools as FireBug, WebKit and Drosera. GNU/Linux (Ubuntu 8.40 - 10.04) operating system is used for the software implementation and deployment.

The requirements for the deployment of the Web-based CNC code editor for gear shaper cutters are summarised in Table 7.5.

7.7 User interface

After a gear shaper cutter design is finished and the automatically generated CNC program is saved to the manufacturing database, CNC operator can access the corresponding CNC file and either directly load it in a CNC machine or modify it using CNC code editor. User interface of the editor is presented in Figure 7.10.

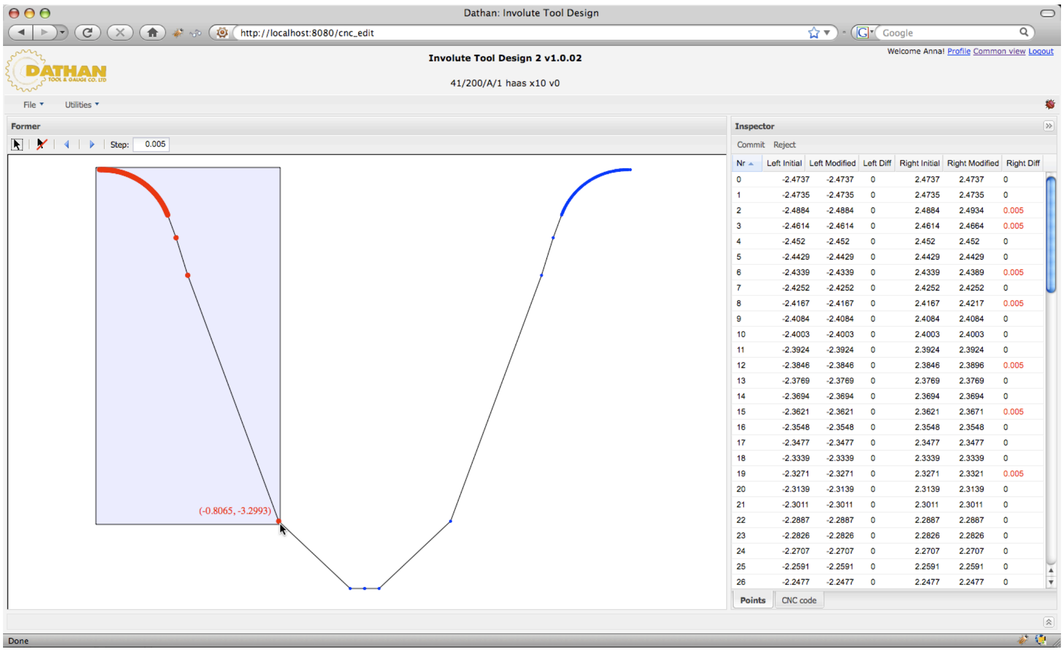


Figure 7.10: CNC code editor. Separate points or a range of points defining tool path could be selected and modified either by hovering over or dragging a selection frame directly on the drawing or by editing coordinates in the list

Main screen of the Web-based CNC code editor has drawing area in its central part. The drawing has interactive tools, which allow selecting separate points or areas on the profile and shift them by a certain amount of inches using controls located on the drawing toolbar. Menu, current CNC file reference and options for modifying user profile data and session termination are located at the top of the screen. Right expandable panel contains two tabs with CNC code and a table of point coordinates and modified values. All unsaved modifications in the table are shown in red colour. The user has an option to define desired coordinate for any point on the profile by directly editing the content of corresponding cell in the table. To apply or reject changes made in the table to the graph, the user is provided with two buttons located at the top of the table: *Commit* and *Reject*.

CNC code incorporating all modifications could be viewed on the *CNC code* tab in the right expandable panel. The user can make extra modifications in the the CNC code, but those changes will not be recognised by the editor to take an effect on the graph.

7.8 Challenges and solutions

7.8.1 Ellipse approximation for the CNC code

The geometry of the former, used to produce gear shaper cutters, often incorporates curved sections, determined by the design modifications, such as root radius, fillet radius, curved root rise and tip relief. As the profile of the former is a subject to the correction factor, all radial sections are converted to elliptical. Due to the limitations of the G programming language ellipses are often hard to program accurately.

The easiest solution would be to use the profile approximation with a polyline to produce the CNC program, as the polyline is already used for the design computations. But in this case to produce smooth curved surfaces on the workpiece the polyline approximation would require high points density on the profile. Unfortunately, due to the nature of the cutting process the manufacturing of a workpiece using this kind of a CNC program is associated with certain practical problems and result in lower quality of the produced tools.

Therefore all the curved sections have to be approximated with circular arcs in order to enable using arc move instructions (G02 and G03 codes) for the accurate manufacturing of elliptical sections on the former profile. As the problem with the polyline approximation has been identified late in the case study development, particularly during the testing of the generated CNC programs on real production examples, the remaining time reserve has been utilised for the implementation of the solution.

Similar to the tooth profile error problem, resolving ellipse approximation task required from the development team additional knowledge and skills, in particular applied geometry and mathematical expertise.

7.8.2 Tracking geometry changes in large CNC programs

The initial design of the graphical user interface for the Web-based CNC code editor implied the user interaction with the visualisation model of the diamond tool path and the interaction with the CNC program code itself. The prototype of the CNC code editor indicated, that making and tracking changes in large sets of coordinates for customised former profiles is not very convenient, as the graphical model could not effectively provide convenient numerical evaluation of the profile geometry changes, neither could do the textual representation of the CNC code.

It was agreed, that a table displaying tool path coordinates and their changes would serve best for the specified task. The implementation of this feature, simple at the first sight, actually had a significant impact on the underlying business logic as it affected the design of the graphical user interface, the associated data flow and the procedures for the synchronisation of the displayed information.

The implemented table of the former profile coordinates uses the data from the design database (if available) to calculate the modifications against the originally designed former geometry. As the point ordinate is never a subject to modifications, the table displays only the original abscissa of each modified point, its current value and the difference between them. Colour coding is used to highlight any changes to the coordinates of the former profile points and the dedicated controls enable committing or rejecting the changes, that the user did in the table manually. The data displayed in the table are updated automatically upon the changes made to the visual model or CNC code.

The problem of tracking the former geometry changes in large CNC programs demonstrates how the initial prototyping facilitated eliminating of an important flaw in usability of the Web-based CNC code editor for GSCs. Thus, the design upstream helped to reduce the amount of re-work, that would be necessary to do in the case of the late discovery of these issues.

7.9 Impact of the development of Web-based CNC code editor

The effects from the development of the Web-based CNC code editor for gear shaper cutter manufacturing programs include the following:

- Improved effectiveness of CNC operators' work;
- Improved application of former inspection data;
- Improved quality of the produced tools;
- Increased production volumes.

The Web-based nature of the editor and integration with inspection enables new possibilities for full cycle manufacturing and quality analysis.

7.10 Summary

In this Chapter the proposed methodology for Web-based CAD/CAM software development was applied for the development of a Web-based CNC code editor. The Chapter provides a short description of the case study and software concept, including its purpose, relevant business processes and workflow, covers the process of user requirements elicitation, software design, architecture, implemented technologies and user interface. The case study contains an overview of the development process, demonstrates the use of the proposed approach to planning and describes real-life application of the mechanism for reserve management, introduced in Section 5.3.

Like in the first case study, Web-based CNC code development encountered some challenges, which provided possibilities to confirm the effectiveness of the methodology and enabled the development of approaches for eliminating uncertainty, upstream optimisation of software architecture, exploratory development and process management. In particular, the effectiveness of initial prototyping for eliminating flaws in the usability of the Web-based CNC code editor for GSCs and reducing the amount of associated re-work is demonstrated by the example of solving the problem of tracking the geometry changes in large CNC programs.

The effects from the development of the Web-based CNC code editor for gear shaper cutter manufacturing programs are reviewed at the end of the Chapter and include improvements in effectiveness of CNC operators' work, application of inspection data, quality of the produced tools and increase if production volumes. The Web-based implementation of the CNC code editor and integration with inspection enables new possibilities for full cycle manufacturing and quality analysis.

Chapter 6 and Chapter 7 demonstrate application of the proposed methodology for the development of two different case studies: a Web-based CAD/CAM system for involute spur gear shaper cutters and a Web-based CNC code editor. This way the applicability of the methodology was confirmed and many of its features were put to test in the circumstances of real development.

During the development of both case studies, some tasks required the development team to have certain knowledge and skills, for example, deep knowledge of the gearing geometry, applied geometry and mathematical expertise, strong analytical skills and ability to effectively communicate with other team members. Therefore, to enable enhanced team productivity, knowledge exchange and collaborative problem solving in a team with diverse expertise, the necessary infrastructures for communication, team collaboration, negotiation, outcome evaluation and revision were provided.

The novel approach for planning based on time reserve management and task prioritisation was applied for the development of both case studies and ensured the flexibility required for overcoming uncertainty, handling changing user requirements and exploratory development while maintaining the main focus on the case study objectives.

Chapter 8

Validation

In this Chapter, effectiveness of the proposed methodology for Web-based CAD/CAM software development is checked through validation, evaluation and analysis of case study results. The validity of results produced by the Web-based CAD/CAM systems are validated using three examples of real usage for each case study. Then, the systems are evaluated through industrial use to prove readiness of the software for use in production and questionnaires are conducted to obtain industrial feedback and identify advantages of using the Web-based systems over similar desktop software. Finally, advantages and limitations of the developed software products are analysed to identify strengths and weaknesses of the developed Web-based CAD/CAM software. A summary about the validation results concludes the Chapter.

8.1 CAD/CAM system for spur involute gear shaper cutters

8.1.1 Real design examples

Three real design cases are presented to examine the system in-depth. The design cases were selected in such a way, that allows to cover as different design workflows as completely as possible. For this reason the first selected design incorporates a complicated design workflow of a cutter with some key tooth profile modifications defined from the given external master gear details. The second design describes a disk type tool with custom teeth pattern and is defined from cutter details. The third design case is defined from internal master gear details and includes a check for trimming. The first design case is reviewed in this section, while the other two are located in Appendix B and Appendix C.

Table 8.1: Cutter details

Parameter	Value
No of teeth	20
Fillet type	Round
Height of fillet	0.72 mm
Oversize	1.7643 mm
Clearance	0.61 mm
Life for graphs	16 mm
Top rake	6.12°
Front rake	5°

To produce a complete design of a gear shaper cutter it is required to define tooth geometry, select an appropriate manufacturing process, specify the shape of the tool body and define any customisations to the pattern of teeth and spaces on the tool face, and finally generate manufacturing instructions for a CNC machine and produce technical drawings and documents with the detailed product information. In the selected design case of a spur gear shaper cutter for cutting given external master gear the tooth geometry is defined by entering master gear parameters and specifying some cutter design details, such as the number of teeth on the cutter, top and front rakes, fillet details, clearance and design oversize. The details of the master gear and the cutter are provided in Table 8.2 and Table 8.1 accordingly.

The gear required to cut is an external spur gear with the outer diameter of 253 mm and 100 small teeth. The gear tooth profile is shown in Figure 8.1, it has a chamfer near the top of the tooth, which is cut at 55° angle and a nice round fillet near the tooth root. The design is calculated for an oversize of 1.7643 mm and allows the clearance of 0.61mm between the bottom of cutter tooth space and the crest of the gear tooth when cutting the gear.

When the cutter profile is defined through the design master gear there is a chance to get a cutter profile with poor characteristics. In the selected design case the cutter tooth profile calculated directly from the master gear dimensions would look like it is shown in Figure 8.2a.

The properties of the gear tooth profile, produced by the obtained cutter profile, are examined to ensure compliance with the specification and quality. The capability for smooth transmission of power between meshing gears is enabled by the involute tooth form, therefore it is important to ensure that the gear tooth profile does not deviate from the true involute line on the most part of the tooth flank. For this purpose involute

Table 8.2: External master gear details

Parameter	**Value**
Gear type	External
MOD	2.5
Pressure angle	20°
No of teeth	100
Major diameter max	253 mm
Minor diameter max	242.638 mm
Minor diameter min	242.538 mm
Circular tooth thickness	3.50106 mm
Max measurement over D=4 mm pins	253.808 mm
Min measurement over D=4 mm pins	253.661 mm
Chamfer parameters	
Chamfer diameter max	252.6 mm
Chamfer diameter min	252.5 mm
Chamfer angle	55°
Redesign	
Redesign type	Simple
Pressure angle	18.4°
Manual root diameter	242.61084 mm
Manual chamfer diameter	252.58046 mm

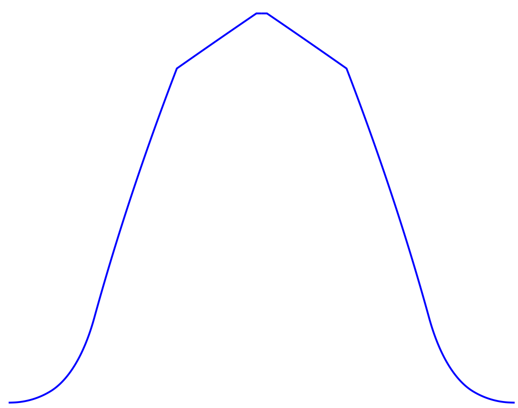
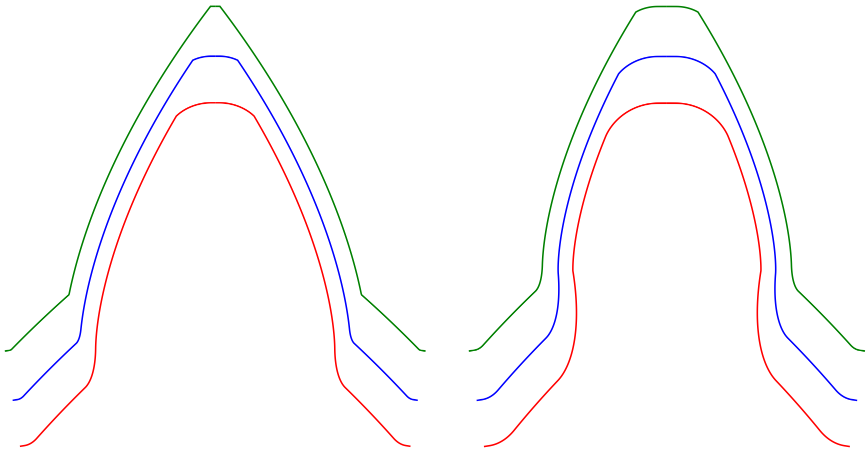


Figure 8.1: Design master gear tooth profile

deviation graph is used, that presents tangental deviation of the tooth profile from the theoretical ideal involute curve on the tooth profile (see Figure 8.3).

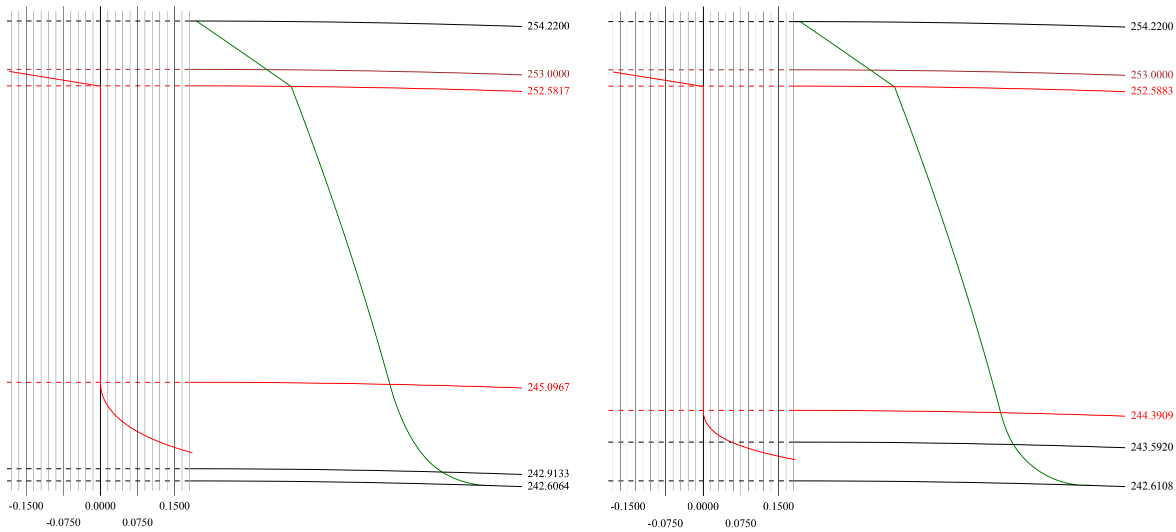
In this particular design case the gear tooth produced by the cutter defined directly from the master gear details has a proper involute profile starting from the diameter of 245.0967 mm up to the diameter of 252.5817 mm, as shown in Figure 8.2a. The true involute portion of the gear tooth profile could be extended by redefining gear at the pressure angle 25°. The redesigned cutter will have a different tooth shape, which is presented in Figure 8.2b, and will be able to cut the gear with smaller fillet and extended



(a) Cutter tooth profile defined by the design master gear

(b) Cutter tooth profile defined by the master gear redesigned at pressure angle 25°

Figure 8.2: Cutter tooth profile design from master gear data and redesign at different pressure angle. Nominal tooth profile is shown in blue, new cutter in green and red is the profile at the end of the tool life (in this case at the distance of 16 mm from the front of the tool).



(a) Involute deviation graph for the design master gear

(b) Involute deviation graph for the redesigned master gear

Figure 8.3: Involute deviation graph, showing tangential deviation (red curve on the left) of the tooth profile (green) from the theoretical ideal involute curve on the master gear. Horizontal marks show tooth profile curve breaks between inside and outside diameters. Diameters that correspond to the starting and ending points of gear tooth flank profile matching true involute are shown in red colour

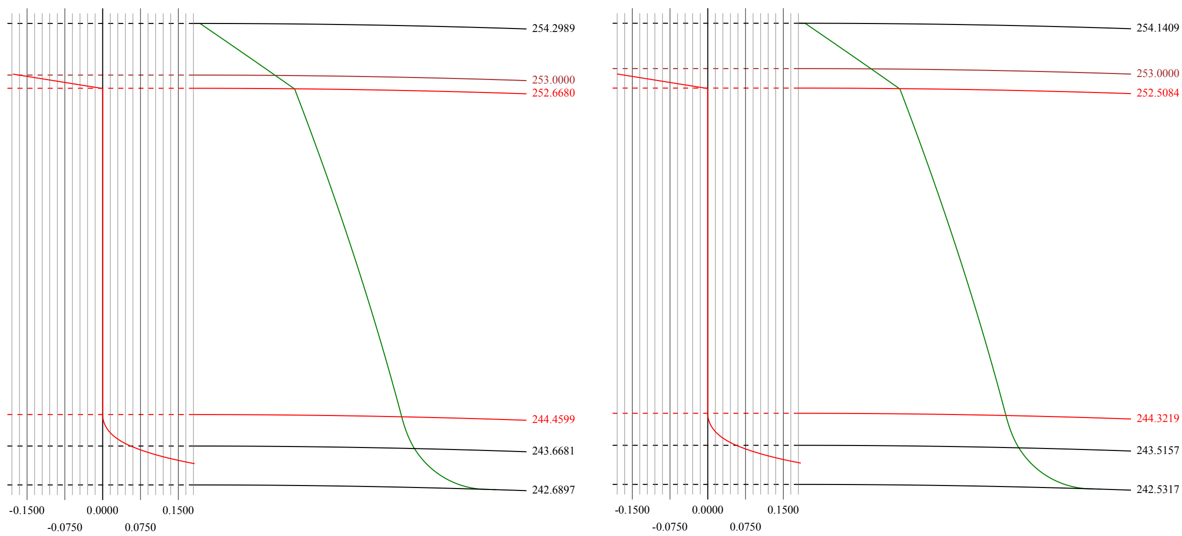
involute profile starting from the diameter of 244.3909 mm up to the diameter of 252.5883 mm, as shown in Figure 8.3b.

In some cases one and the same cutter is used to produce several different gears. In the reviewed design case the cutter is optimised to cut gears with variable circular tooth thickness in the range between 3.47452 mm and 3.52759 mm. For this purpose two auxiliary gears with maximum and minimum specified circular tooth thickness are reviewed. For the convenience those are called MAX gear and MIN gear accordingly. The parameters of all three gears covered by this design case are listed in Table 8.3.

Table 8.3: Multiple gears

Parameter	**MAX gear**	**MIN gear**	**Master gear**
Gear type	External	External	External
No of teeth	100	100	100
Major diameter max	253 mm	253 mm	253 mm
Minor diameter max	242.638 mm	242.638 mm	242.638 mm
Minor diameter min	242.538 mm	242.583 mm	242.538 mm
Circular tooth thickness	3.52759 mm	3.47452 mm	3.50106 mm

The tooth profiles, that the designed cutter produces on the secondary gears with the maximum and minimum circular tooth thickness, are presented in Figure 8.4, provided with the involute deviation graphs to verify the quality of the resulting gear tooth profiles.



(a) Involute deviation graph for the gear with maximum allowed circular tooth thickness

(b) Involute deviation graph for the gear with minimum allowed circular tooth thickness

Figure 8.4: Involute deviation graph for the secondary gears with minimum and maximum allowed circular tooth thickness. The involute deviation graph is shown in red on the left side from the half of the gear tooth profile, which is shown in green

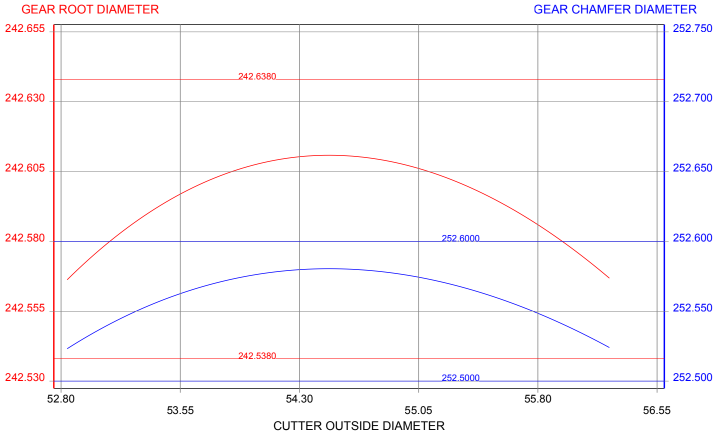
The design of a gear shaper cutter should be also optimised in regards to the numerous re-sharpening operations performed with time. The cutter re-sharpening results in changed cutter tooth profile and leads to changes in the gear root and chamfer diameters. Depending on the variations in the gear root and chamfer diameters as the re-sharpening operations are carried out on the cutter, the designer decides on the boundaries for the cutter outside diameter and can optimise the cutter performance throughout the tool life by redefining gear root and chamfer diameters manually.

In the selected design case the root and chamfer diameters of the design master gear are optimised manually by entering values provided in Table 8.2. Redesigning the gear at the root diameter 242.61084 mm ensures that the gear root diameter is within the defined minimum and maximum boundaries throughout the whole cutter life. Similarly redefining master gear chamfer diameter at 252.58046 mm adjusts the gear chamfer diameter curve between the minimum and maximum boundaries of the chamfer diameter. Figure 8.5 presents variations in the chamfer and root diameters of the design master gear and two auxiliary gears depending on the cutter re-sharpening.

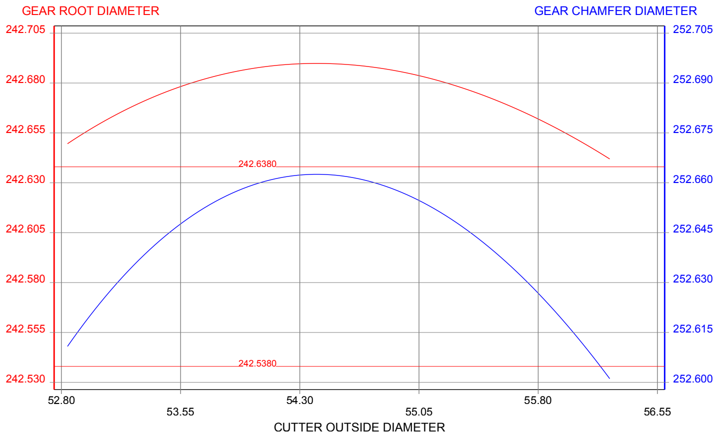
As the manufactured gear shaper cutters undergo the inspection process, some additional cutter tooth parameters are required to be calculated during the design. These parameters are corrected for the top and front rake angles of the gear shaper cutter and include, for example, finished circular tooth thickness, pressure angle at finished outside diameter, nominal and finished outside diameters, side relief angle and others. The calculated parameters for the selected design case are listed in Table 8.4.

The manufacturing section of the CAD/CAM system for gear shaper cutters allows the designer to select the most efficient way to manufacture the product. The manufacturing process of a gear shaper cutter could be split in two phases: the first phase aims to remove as much excessive metal from the cutter blank as possible, and in the second phase a grinding wheel is used to produce the required tooth profile. The grinding provides capabilities for high precision finishing, but is expensive and slow. Therefore a more efficient tool, such as a hob, rack, gear shaper cutter or a milling cutter, is usually used to quickly remove the metal from spaces between cutter teeth before proceeding to grinding.

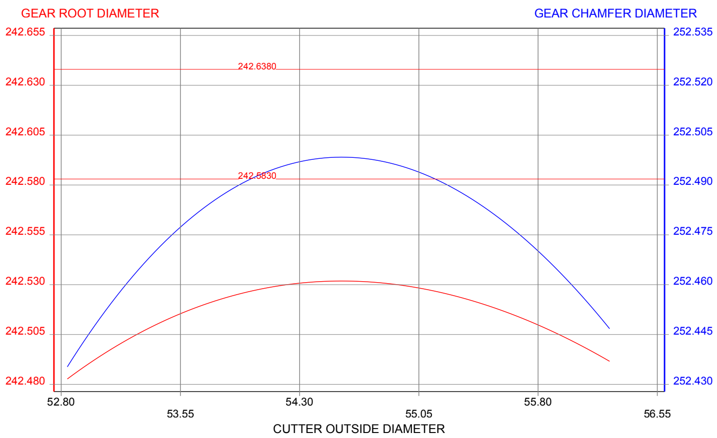
The manufacturing section provides the designer with a set of tools available for gear shaper cutter machining and allows to compare the profile, that a selected tool would cut in the cutter blank, and the required cutter profile. Figure 8.6 presents manufacturing visualisation for the design under review. The blue line in this picture shows the finished profile of the space between the cutter teeth. Grinding allowance is normally set for the cutter tooth flanks, and the green line marks the profile with the allowance for the



(a) Chamfer and root diameter graph for the design master gear



(b) Chamfer and root diameter graph for the gear with maximum allowed circular tooth thickness



(c) Chamfer and root diameter graph for the gear with minimum allowed circular tooth thickness

Figure 8.5: Variations in the chamfer and root diameters of the design master and secondary gears throughout the cutter life

Table 8.4: Design results

Parameter	Value
Addendum	2.48481 mm
Dedendum	2.70977 mm
Finished circular tooth thickness	4.05859 mm
Pressure angle at finished OD	33.35476°
Former tip thickness	2.09760 mm
Tip thickness at finished OD	0.97787 mm
Nominal outside diameter	54.48571 mm
Finished outside diameter	56.25001 mm
Former pressure angle	18.6614°
Corrected pressure angle	18.56235°
Correction factor	0.98497
Side relief angle	2.06206°
Radius of fillet	1.05209 mm
Height of fillet	0.72 mm
Max height of fillet	0.86277 mm
Min height of fillet	0.59737 mm
Base circle diameter	46.94015 mm
Pitch circle diameter	49.51609 mm

designer. Finally the yellow line shows the profile cut in the cutter blank by the selected manufacturing tool.

In this particular design a gear shaper cutter with identical base pitch and cutter depth 6.627 mm has been selected to manufacture the designed cutter. As seen in Figure 8.6, the profile produced by the selected tool comes close to the profile with flank allowance. And despite it has rather big clearance for topping, using this tool would leave very little of excessive metal in spaces between the cutter teeth, thus significantly improving the efficiency of the whole manufacturing process.

Referring to manufacturing details for the chosen design listed in Table 8.5, the clearance for thickness between the selected manufacturing tool and the designed cutter is only 0.815 mm. The depth of the manufacturing cutter is 6.627 mm depth and the clearance for the cutter to cut is set to 0.61 mm for better results.

As the design does not require any customisations to the pattern of teeth and spaces on the cutter face, it is only remained to choose the shape for the cutter body and define its parameters. The CAD/CAM system provides the designer with a range of cutter bodies with standard dimensions for defined body shape, type and bore size, that can be further specified in detail or modified if required. Alternatively the user can manually define parameters to construct a customised body from scratch. The body details for the selected design case are listed in Table 8.6.

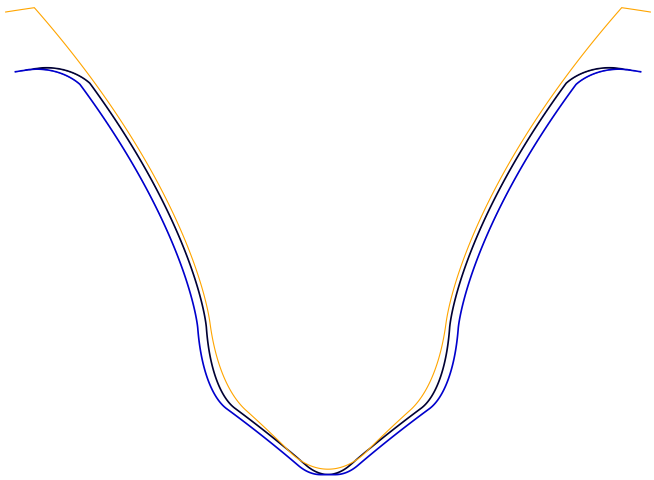


Figure 8.6: Manufacturing visualisation, showing finished cutter tooth space profile (blue), cutter profile with allowance on flanks (black) and the profile cut in the cutter blank by the selected manufacturing tool (yellow)

Table 8.5: Manufacturing details

Parameter	**Value**
Tool	Shaper
Type	NC
Base pitch	7.38 mm
Clearance for thickness	0.815 mm
Clearance for topping	1.432 mm
Cutter depth	6.627 mm
Clearance for cutter to cut	0.61 mm
Allowance/Flank	0.117 mm
Design base pitch	7.38 mm
Ideal clearance	0.99 mm
Actual clearance	0.61 mm

The created gear shaper cutter is designed with a shank type body, having 72 mm long neck, No. 4 Morse fitting and 16 mm SI thread. There is also specified tool face recess with 25.4 mm diameter and 9.526 mm depth, and also a standard large centre hole on the face recess surface. The cutter teeth are defined to be 20 mm long with 5 mm chamfer cut at 30° angle at the tooth back end. The back side of the tooth has 7 mm radius. A drawing of the specified cutter body is shown in Figure 8.7.

Upon the end of the design technical drawings, job ticket and CNC code can be automatically generated. The CNC code instructions and documents produced for the reviewed design case could be found in Appendix A.

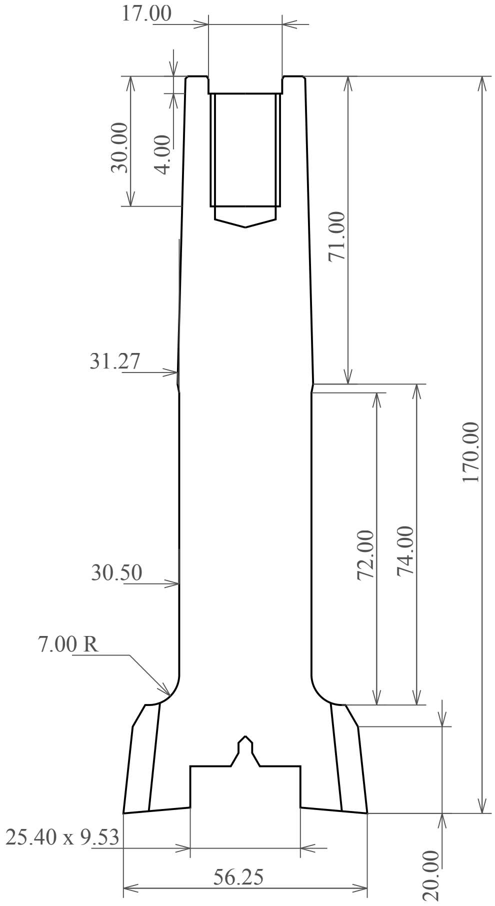


Figure 8.7: Tool body drawing

8.1.2 System evaluation through industrial use

The Web-based CAD/CAM system for gear shaper cutters was examined in production on a large set of real designs. The total number of designs produced using the Web-based system comes to 1086. To encompass wide range of requirements the examination reflects different aspects of gear shaper cutter design, including input parameters defining design workflow, gear type, tooth profile modifications and specification of tool body. Table 8.7 presents decomposition of the examination set in accordance with the specification of tool design, as well as provides the number of designs created for each specification.

As it was described in section 6.2.1 (see Figure 6.2) cutter tooth profile can be defined through gear and cutter parameters or through cutter parameters only. When working out the design from gear and cutter parameters, the cutter is defined through the details of master gear, that is supposed to be cut by this cutter. The number of designs examining master gear workflow is 889. Defining the cutter from cutter parameters only allows for an option to follow a standard cutter design or specify a custom cutter. The number

Table 8.6: Tool body details

Parameter	**Value**
Body shape	Shank
Tooth length	20 mm
Fitting type	No. 4 Morse
Thread type	16 mm SI
Tooth chamfer length	5 mm
Tooth chamfer angle	30°
Tooth back radius	7 mm
Neck diameter	30.5 mm
Neck length	72 mm
Ramp length	2 mm
Fitting length	71 mm
Fitting top diameter	31.27 mm
Taper angle	1.4876 mm
Thread length	30 mm
Thread diameter	16 mm
Thread lead diameter	17 mm
Tapping drill size	14
Face recess	
Face recess diameter	25.4 mm
Fase recess depth	9.525 mm

of examined standard cutter designs comes to 53. The significantly greater number of custom cutter designs, which is 1033, is governed by the business specialisation (note also, that custom cutters may be defined through master gear parameters).

When designing a cutter based on the master gear parameters, the quality of the tool may be improved through redesign by adjusting some of the design parameters. 567 designs involving redesign workflow were created during the use of the Web-based CAD/CAM system in production so far.

Besides that, the same cutter can be used to produce multiple different gears and the new Web-based CAD/CAM system for GSCs supports this kind of design workflow as well. The examination set of cutters with multiple gears encompasses 271 designs.

Then the design calculations greatly depend on the type of the gear, that the cutter will be able to produce. The system was examined on designs of cutters for both internal (622 designs) and external (340 designs) gears.

As the tooth profile modifications make up the most essential part of the design process, Table 8.7 presents examination set breakdown by every possible tooth profile modification. There were examined 214 designs of cutters with chamfering, 99 designs with specified cutter tip rise of gear root relief, 22 designs with specified cutter root rise

Table 8.7: Real designs produced using the Web-based CAD/CAM system

	Specification	**Number of designs**
Input paramaters	Standard cutter	53
	Custom cutter	1033
	Master gear	889
	Redesign	567
	Multiple gears	271
Gear type	Internal	622
	External	340
Tooth profile modifications	Chamfering	214
	Cutter tip rise/ gear root relief	99
	Cutter root rise/ gear tip relief	22
	Cutter protuberance	3
	Gear pre-finishing	102
Tool body	Disk	346
	Shank	419
	EBB	164
	Hub	1
Generated CNC programs	Haas	598
	Diaform	340

or gear tip relief, 3 designs with protuberance on the cutter tooth and 102 designs with specified gear pre-finishing.

Depending on the specified tool body shape, designs could be divided in four groups: disk type tools, shanks, tools with extended back body and hubs. Total number of designs with specified tool body shape is 930. Note, that the total number of designs for some of the aspects may differ from the total number of produced designs, because some of the designs are created to check the gearing geometry and therefore may lack specification of manufacturing details and particularly specification of tool body.

For the examination of CAM portion of the Web-based CAD/CAM system for GSCs, a total number of 938 CNC programs were generated, including 598 haas CNC programs and 340 diaform CNC programs.

During the examination of the Web-based CAD/CAM system for GSCs in production, there were discovered and resolved 142 issues. Table 8.8 shows breakdown of issues by application component and issue type. The most prevalent issues were associated with

Table 8.8: Resolved issues discovered in the Web-based CAD/CAM system for GSCs

Component	Bugs	Enhancement	Proposal	Task	Total
CNC code	9	0	0	1	10
Body shape	8	1	0	0	9
Chamfer root graph	3	0	0	0	3
Common graph	8	1	0	0	9
Design DB	5	0	0	0	5
Gearing calculations	9	1	0	0	10
Manufacturing	7	0	0	0	7
Multigraph	21	0	0	0	21
PDF export	39	1	1	0	41
Undefined	25	2	0	0	27
All components	134	6	1	1	**142**

software errors and the largest amount from all issues fall on the most complex and uncertain parts of the system – multigraph and PDF export.

Table 8.9: Priority of issues in the Web-based CAD/CAM system for GSCs

Priority	Number of issues
Trivial	13
Minor	61
Major	53
Critical	8
Blocker	7

As it follows from Table 8.9, which sums up the number of issues by their priority, the most of the issues have had their priority from trivial to major, with a few critical issues and blockers.

8.1.3 Industrial feedback

The estimation of benefits from using the Web-based CAD/CAM system for GSCs over similar desktop applications is issued from the specialist opinions. To ensure, that testimonials of industry representatives are based on real experience of using the system for solving business tasks and thus correctly reflect the situation in the company, the feedback was obtained one year after the Web-based CAD/CAM system for GSCs has been launched in production.

Two company's representatives who are the most competent in design and manufacture of gear shaper cutters were asked to fill in a questionnaire and compare the Web-based CAD/CAM system with the previously used desktop CAD/CAM software by providing estimates from 1 to 5 for a number of questions, where 1 is the lowest mark and 5 in the highest mark. The questionnaire is presented in Appendix G and contains questions about software usability, interactivity, reliability, collaboration and scalability. At the end of the questionnaire respondents are asked to describe the impact on company's operations and the performance enhancements for the company due to the implementation of the Web-based CAD/CAM system for GSCs.

Average estimates of Web-based and desktop CAD/CAM software usability, interactivity, reliability, collaboration and scalability provided by the respondents are marked on a radar diagram, which is presented in Figure 8.8. Each axis of this radar diagram corresponds to one of the mentioned software characteristics. As it follows from the Figure 8.8, the developed Web-based CAD/CAM system was given usability, interactivity and reliability estimates equally good or slightly better than those for desktop software. Apart from that the graph corresponding to the Web-based CAD/CAM system is more balanced, that the graph of the desktop CAD/CAM system, as it has also high marks for collaboration and scalability.

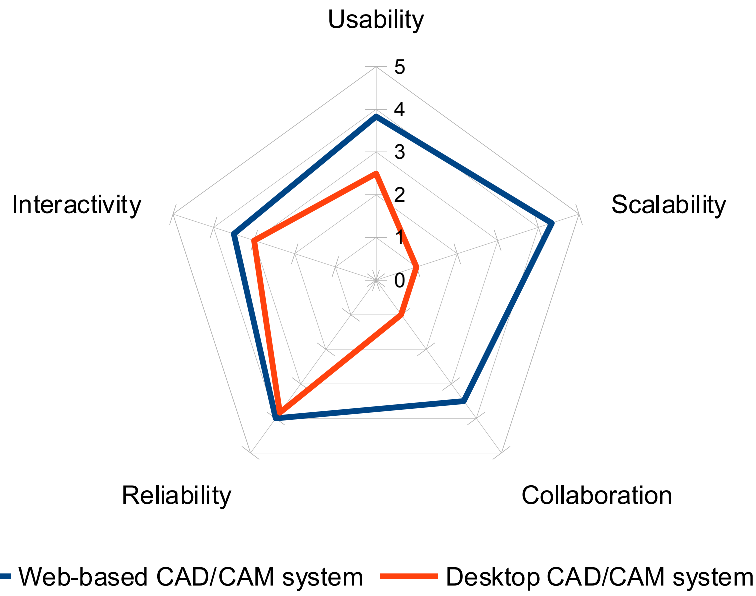


Figure 8.8: Evaluation of the Web-based CAD/CAM system compared to the desktop CAD/CAM system based on questionnaire results

Regarding the impact on company's operations from the implementation of the Web-based CAD/CAM system for GSCs, first of all industrial representatives point out the capability of distributed access to the CAD/CAM system governed by its Web-based implementation: "Freedom in geographical location allow design staff and foreign agents to define customer requirements and solve problems more efficiently". The capability of remote access to the CAD/CAM system is important to the company considering the geographical distribution of customers and business partners as this enables new opportunities for improving business operations.

The interviewees also mentioned the long-term implications from using Web-based technology due to the increased protection against changes in workstation software and hardware. Also since the Web-based CAD/CAM system uses a centralised server for data storage and processing it could be easier interfaced with other company operations.

The new Web-based CAD/CAM system has been characterised as easy to learn and use, which allows new design staff to be quickly productive. The company has reported design lead time reduced twice compared to the previously used desktop software, thus improving response time and helping with on time delivery of product.

8.1.4 Analysis of advantages and limitations

Analysis of advantages and limitations of the developed Web-based CAD/CAM system for GSCs is provided to outline strengths and weaknesses of the software solution, implemented by following the proposed software development methodology.

The Web-based CAD/CAM system for GSCs could be characterised as follows:

- The system is based entirely on Web technologies and could be used from any place and platform (even a smartphone) via the Internet;
- Only a modern Web browser is required for enabling full functionality of the system without the need to install additional tools;
- The system does not require software installation, maintenance, configuration or updating on the client side;
- Application has centralised architecture;
- Developed solution is software and hardware platform independent;
- Modular user interface based on expandable panels ensures effective management of displayed information, supports personalisation and is highly adjustable and scalable;
- The system is flexible, easy to extend and modify;
- Offered Web-based CAD/CAM system design ensures robust, crash-proof and reliable program performance;

- The CAD/CAM system could be easily integrated with other software, for example, customer relationship management system or product inspection software;
- The system does not use proprietary components and is delivered with open source, simplifying further program maintenance.

These key characteristics of the Web-based CAD/CAM system enable a range of advantages provided by the Web-based implementation, which along with the limitations of the CAD/CAM system are listed in Table 8.10. The advantages provided by the developed CAD/CAM system for GSCs are firstly associated with its Web-based implementation, establishing distributed assess over the Internet and collaboration capabilities and facilitating CAD/CAM integration with other company's operations; secondly they are associated with the improved gearing model computations, extended functionality and better support for business processes (compared to previously used desktop CAD/CAM software); and thirdly the advantages are associated with the application design ensuring scalability, flexibility, extensibility and maintainability of the system. Hardware and software platform independence enable wide choice of the deployment platform, whereas application centralisation simplifies operations associated with the deployment, backup and administration. The system also has been designed to provide robust, crash-proof and reliable program performance, establishing modular user interface, which is easy to learn and use.

Table 8.10: Advantages and limitations of the Web-based CAD/CAM system for GSCs

Advantages	**Limitations**
• Distributed access. • Collaboration capabilities. • Scalable architecture. • Wide choice of deployment platform. • Easy deployment, backup and administration. • Flexible, easy to extend and modify. • Good maintainability. • Capabilities for easy integration with other company operations. • Robust, crash-proof and reliable program performance. • Easy to learn and use.	• All technical drawings are generated in 2D. • System performance depends on the quality of network connection between the server and the client. • Current implementation does not support file formats, compatible with other popular CAD software packages.

The limitations of the developed Web-based CAD/CAM system on the one hand are governed by the Web-based architecture, making system performance dependant on the quality of network connection between the server and the client; and on the other hand are related to the application context and current business requirements, as there is no real need for 3D representation of technical drawings, as well as for the support of file formats, compatible with other popular CAD software packages, although, this could be included in future releases of the Web-based CAD/CAM system.

8.2 Web-based CNC code editor

8.2.1 Usage example

To validate the results produced by the Web-based CNC code editor the results produced when making modifications to three randomly selected CNC programs are examined. The listings for examined CNC programs is located in Appendix D, Appendix E and Appendix F.

This section reviews the use case of modifying the first selected CNC program in detail. The CNC program presented in Appendix D.1 contains Haas instructions for manufacturing a former, intended to produce a gear shaper cutter for an internal gear with curved tip relief of 0.004 mm.

During the inspection of the former it was discovered, that the machined contour does not conform with the required geometry specification. One side of the former deviates slightly near the tip, thus a tool path correction is required to compensate the machining inaccuracy. The machining errors may be caused by various factors, including geometric errors, thermally induced errors, load induced errors and other, such as vibration, deformation, fixturing and controller errors. The offline error prediction process, implemented in the company, accounts for most of the sources of machining errors and involves direct post-process measurement using a Coordinate Measurement Machine (CMM).

After the estimation of the machining errors the tool path is modified to account for those errors. The Web-based CNC code editor provides capabilities for quick alteration of the manufactured contour by manipulating with coordinates of a selected part of the former profile, as shown in Figure 8.9. As the ordinate error could be neglected, only abscissa errors are the subject for tool path correction. For the chosen example of the Web-based CNC code editor usage Table 8.11 presents initial, modified and the difference between these two abscissae of the left and right sides of the former.

As it was mentioned above, the inspection revealed that there is about 0.005 mm deviation along the abscissa axis between the former geometry specification and the

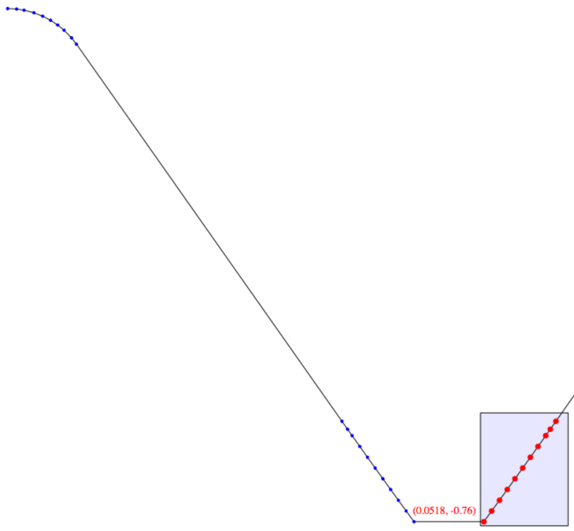


Figure 8.9: Drawing of the profile and selection of coordinates for tool path correction

Table 8.11: Set of coordinates for the example profile

Nr	**Left Initial**	**Left Modified**	**Left Diff**	**Right Initial**	**Right Modified**	**Right Diff**
0	-0.6538	-0.6538	0	0.6538	0.6538	0
1	-0.6404	-0.6404	0	0.6404	0.6404	0
2	-0.6295	-0.6295	0	0.6295	0.6295	0
3	-0.6149	-0.6149	0	0.6149	0.6149	0
4	-0.6019	-0.6019	0	0.6019	0.6019	0
5	-0.5902	-0.5902	0	0.5902	0.5902	0
6	-0.5797	-0.5797	0	0.5797	0.5797	0
7	-0.5702	-0.5702	0	0.5702	0.5702	0
8	-0.5591	-0.5591	0	0.5591	0.5591	0
9	-0.5519	-0.5519	0	0.5519	0.5519	0
10	-0.1585	-0.1585	0	0.1585	0.1583	-0.0002
11	-0.1502	-0.1502	0	0.1502	0.15	-0.0002
12	-0.1434	-0.1434	0	0.1434	0.1432	-0.0002
13	-0.132	-0.132	0	0.132	0.1318	-0.0002
14	-0.1207	-0.1207	0	0.1207	0.1205	-0.0002
15	-0.1093	-0.1093	0	0.1093	0.1091	-0.0002
16	-0.0978	-0.0978	0	0.0978	0.0976	-0.0002
17	-0.0864	-0.0864	0	0.0864	0.0862	-0.0002
18	-0.0749	-0.0749	0	0.0749	0.0747	-0.0002
19	-0.0633	-0.0633	0	0.0633	0.0631	-0.0002
20	-0.0518	-0.0518	0	0.0518	0.0516	-0.0002
21	0	0	0	0	0	0

profile of the machined part, produced by the selected CNC program, which means that the tool path should be corrected by 0.0002 inches (all the dimensions in the CNC program are in imperial measurement system). The corrected CNC instructions are then sent to the machine for finishing the part. The CNC program, modified with respect to the estimated error, could be found in Appendix D.2.

8.2.2 System evaluation through industrial use

To examine the Web-based CNC code editor it was used to edit a large set of different type CNC programs, which contain instructions for producing tooth formers containing all possible tooth profile modifications. Referring to Table 8.12, with the use of the Web-based CNC code editor there have been inspected and modified 165 haas CNC programs and 128 diaform CNC programs.

Table 8.12: CNC programs modified using the Web-based CNC code editor

CNC program format	**Number of programs**
Haas	165
Diaform	128

8.2.3 Industrial feedback

The estimation of advantages from the Web-based CNC code editor development is based on the testimonials of two competent specialists from industry, who have experience of using both applications for editing CNC programs and solving manufacturing tasks. The selected specialists were asked to fill in a questionnaire compare the Web-based CNC code editor with desktop applications used by these specialists for CNC code editing previously, namely ESPRIT and Notepad for Microsoft Windows.

The questionnaire is presented in Appendix H and asks to give estimates from 1 to 5 for a number of questions regarding software usability, interactivity, reliability, collaboration and scalability. At the end of the questionnaire respondents are asked to describe the impact on company's operations and the performance enhancements for the company due to the implementation of the Web-based CNC code editor.

Figure 8.10 summarises the questionnaire results in a form of a radar diagram, where average estimates of usability, interactivity, reliability, collaboration and scalability are marked on five corresponding axis, this way constructing two graphs for Web-based and desktop CNC code editing software. Like in the case with the Web-based CAD/CAM system (see Figure 8.10), it could be noted that the graph corresponding to the Web-based CNC code editor is more balanced, than the graph of the desktop editors. The developed Web-based CNC editor was given usability, interactivity and reliability estimates equally good or slightly better than those for desktop software, and unlike the desktop editors the Web-based application received also high collaboration and scalability estimates.

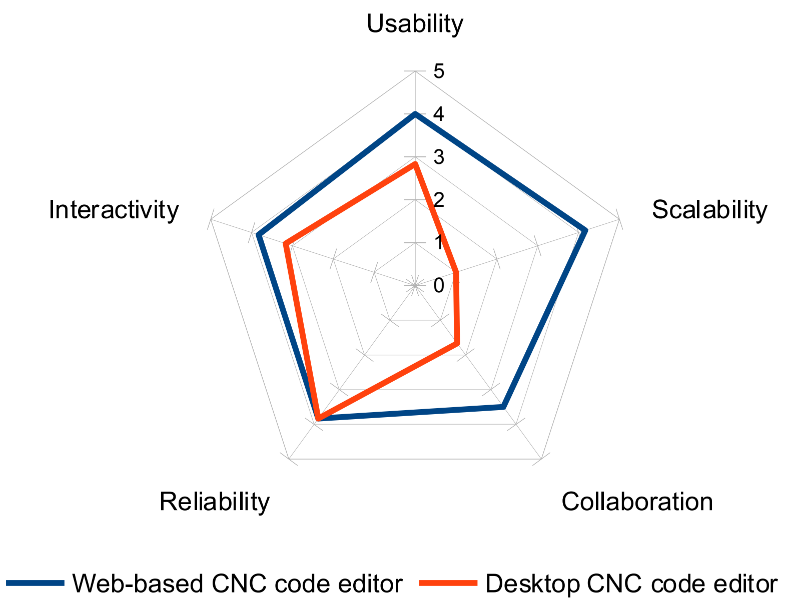


Figure 8.10: Evaluation of the Web-based CNC code editor compared to the desktop software based on questionnaire results

For the estimation of the impact on company's operations from the Web-based CNC code editor implementation the respondents were asked to describe the Web-based CNC code editor workflow, estimate the impact on company's operations and the performance enhancements for the company due to the implementation of the Web-based CNC code editor for gear shaper cutters.

The users of the Web-based CNC code editor remark that it takes less time to alter the former and the number of mistakes made when altering a CNC former is significantly reduced compared to the method used previously. The visual representation of the former shape and the controls allowing to select what needs to change and by what amount make the process of altering the former very easy and straightforward.

The Web-based implementation of the CNC code editor also made possible the integration with CAD/CAM and inspection operations and design database, thus essentially improving internal business processes.

8.2.4 Analysis of advantages and limitations

Similar to the Web-based CAD/CAM system for GSCs developed in the first case study, the Web-based CNC code editor provides advantages governed by the factors, that may be divided in three groups:

- Web-based implementation;
- Improved support for underlying business processes;
- Application design addressing key software development concerns.

Probably, the most important advantage gained from the Web-based implementation of the CNC code editor is the capability to link the process of altering CNC programs with CAD and inspection operations. Moreover, centralised application architecture significantly simplifies deployment, backup and administration and enables storing and re-using created CNC programs.

Table 8.13: Advantages and limitations of the Web-based CNC code editor

Advantages	**Limitations**
• Linked with CAD and inspection operations. • Enables storing and re-using created CNC programs. • Ensures verification and plotting of the former geometry. • Easy deployment, backup and administration. • Wide choice of deployment platform. • Scalable architecture. • Good maintainability. • Easy to learn and use.	• System performance depends on the quality of network connection between the server and the client. • The editor supports only Haas and Diaform CNC syntax. • Current release supports limited scope of CNC code instructions. • The current release cannot do multi-tooth formers. • 2D visualisation of the former.

The support for associated business processes has been greatly improved by implementing automatic profile corrections based on inspection data, as well as by establishing capabilities for verification and plotting of the former geometry. Software development concerns are addressed to ensure application quality, providing good scalability, maintainability and usability.

Both Advantages and limitations of the Web-based CNC code editor are listed in Table 8.13. The limitations of the Web-based CNC code editor include performance dependence on the quality of network connection between the server and the client, limited support for CNC code instructions and syntax, inability to do CNC programs for multi-tooth formers and visualisation of the tool path in 2D.

8.3 Summary

Validation and evaluation of the proposed methodology for creating Web-based CAD/CAM systems is performed through the case study results. Three use cases for each case study demonstrate in-depth examination of the developed Web-based CAD/CAM applications, thus the validity and reliability of the results, produced by both applications, are checked and software compliance with the requirements specification is confirmed. Both Web-based CAD/CAM applications were examined in production on a large set of real use cases. The evaluation through industrial use confirms that both applications are ready to be used in production and are compliant with the business requirements.

Feedback from industry representatives having real experience of using the evaluated software for solving business tasks indicates advantages from using the Web-based CAD/CAM systems over similar desktop applications, such as the capability of distributed access through the WWW, long-term implications of using Web-based technology due to the increased protection against changes in workstation software and hardware, capabilities for easy integration with other company operations, ease of learning and use. The developed Web-based CAD/CAM system and CNC editor were given usability, interactivity and reliability estimates equally good or slightly better than those for desktop software, and unlike the desktop software the Web-based systems received also high collaboration and scalability estimates.

Analysis of advantages of the developed Web-based CAD/CAM applications outlines strengths and weaknesses of the software solutions that were implemented by following the proposed software development methodology. The developed Web-based CAD/CAM applications provide advantages, that may be divided in three groups:

- Web-based implementation advantages;
- Improved support for underlying business processes;
- Application design addressing key software development concerns.

The limitations of the developed Web-based CAD/CAM software are governed by the Web-based architecture, making system performance dependent on the quality of network connection between the server and the client. Other limitations are related to the application context and current business requirements, such as 2D representation of technical drawings and limited support for file formats and CNC code instructions and syntax, although, these enhancements could be included in future releases.

The Web-based CAD/CAM systems produced during the development of case studies could be characterised as robust, crash-proof and reliable with flexible and extendible design, software and hardware platform independency, open source and browser-based implementation, which simplifies further program maintenance and utilises advantages of Web-based applications to the highest extent.

For the sponsor company, the most significant effects from the development of the Web-based CAD/CAM system for GSCs and Web-based CNC code editor include:

- Significantly reduced design lead time compared to the previously used desktop software;
- Improved response time and on time delivery of products;
- Improved effectiveness of designers' and CNC operators' work;
- Improved quality of the produced cutting tools;
- Increased production volumes;
- Improved accessibility and opportunities for user collaboration and customer involvement in product development process;
- Linked design and manufacturing operations;
- Improved company expertise base and production flexibility.

The listed effects from the development of Web-based CAD/CAM applications and advantages associated with their use in production confirm the initial hypothesis underlying this thesis, namely, that Web-based approach provides significant advantages for CAD/CAM software development over the desktop applications.

Chapter 9

Discussion and conclusions

9.1 Key findings

9.1.1 Previous research

The literature review identifies the possibilities for an efficient way of exploiting state-of-the-art Web technologies in the area of CAD/CAM. An overview of Web technology and its application areas demonstrates the overall readiness of the WWW to serve as base technology for the development of complex industrial software systems such as CAD/CAM applications. The review of past and present software development practices indicates that the creation of Web-based CAD/CAM applications is in line with common trends in software development. The reasons for increasing popularity of Web applications are investigated and arguments in favour of Web-based software development and the possible benefits for the industry from implementation of this kind of systems are stated.

An analysis of publications indicated significant growth of research interests in the fields of Web-based systems, Web-based PDM and Web-based DSSs since 2000. The development of enabling graphical technologies in the late 90's resulted in a splash of researcher interests, which are the most perceptible in the area of Web-based PDM. Web-based DSS research covers many different subject areas and apart from computer science there are many publications in engineering, which all together indicate the potential of the application of Web technology in these areas.

A number of Web-based CAD/CAM solutions described in literature were analysed to summarise the research in this field and investigate the implementation methods and techniques. A relevant classification of this kind of systems has been established and a few conclusions made regarding the approach to the development of industrial Web-based CAD/CAM systems. A scope of associated challenges has been outlined too.

An analysis of the existing Web-based CAD/CAM solutions revealed that almost a half from all these applications are not complete solutions and therefore could not be used within an industrial environment. This finding confirms the claims of other researches about a large number of Web systems in product design and manufacture being developed as proof-of-concept prototypes. It has been discovered that a large part of surveyed CAD/CAM applications were developed using integrated solutions and well known programming technologies, which along with reducing risks associated with novelty and complexity of the project, limits the extent of utilising Web-based technology advantages in terms of flexibility, compatibility, openness and ease of maintenance.

It was also found that despite a plethora of Web-based technologies there is no defined generic approach for the development of Web-based CAD/CAM systems, as well as there is no defined system architecture and development methodology for complex industrial Web-based CAD/CAM application. A review of software development methodologies outlined the possibility for adapting software engineering experiences for creating a methodology for Web-based CAD/CAM systems, which combine features of diverse domains, namely Web applications, specialised CAD/CAM software and science-intensive and complex CSE software.

9.1.2 Rationale and prerequisites for methodology

The thesis mentions characteristics specific to Web-based CAD/CAM software development that make the application of the existing methodologies troublesome. These characteristics are listed (section 4.6) and include ambiguous and changing requirements, which usually contain specific industry-related and often scientifically-intensive information, stretched over time development and continuous evolution of the application, changing technologies and high level of uncertainty, as well as involvement of people with expertise in different fields specifically for the development of this particular software.

Based on these features specific to Web-based CAD/CAM software, a set of challenges in the development of industrial Web-based CAD/CAM applications is identified (section 4.7). To address the identified challenges of Web-based CAD/CAM software, the main prerequisites for the Web-based CAD/CAM software development methodology are formulated (section 4.8). The methodology is required to address challenges governed by the features specific to industrial Web-based CAD/CAM applications and should provide techniques that would enable effective and quality software development.

9.1.3 A methodology for creating Web-based CAD/CAM systems

To describe a software development methodology in an organised and structured manner, a set of relevant concerns was provided to assist a developer in identifying if the established methodology could be applied in a particular project, what benefits it can bring, how to address different software development concerns and what steps to undertake to apply the methodology.

The methodology is described to provide guidance on the development of complex Web-based CAD/CAM systems to industrial quality standards in a time and cost effective manner. Based on the identified challenges specific to Web-based CAD/CAM system development, major considerations are given to address associated issues and support the development process. Subsequently, using these considerations key principles are outlined that need to be followed when developing Web-based CAD/CAM applications. The proposed methodology includes such principles as user involvement, iterative and incremental development process focused on frequent delivery, development prioritisation based on task complexity and relevance to current situation, giving preference to most complex and critical tasks to be developed first, exploratory requirement identification, use of throwaway prototyping, design patterns and refactoring, development planning incorporating time reserves, continuously adjusted effort estimates, efficient and effective communication between all involved parties. Also a novel project development process model is established to enable investigation of the optimal application architecture before getting to the actual incremental development, thus providing capabilities for early exploration of critical design features.

Built on the features of widely used software models, the methodology proposes an optimised model for agile development of Web-based CAD/CAM systems and industrial science-intensive applications and also incorporates several unique features, such as upstreaming design optimisation, minimising the necessity to deal with the resistance to change in incremental development model, supporting research activities and exploratory development, while staying focused on requirements. The work also introduces a novel approach to planning unpredictable software development projects, based on efforts reserved for overcoming uncertainty. The reserve is meant to be gradually used during the project development for uncertainty reduction. An approach was also established to support decision making about the speed of reserve utilisation and the task composition for each iteration, which is based on the task prioritisation and categorisation.

The Web-based CAD/CAM software concept described in this thesis can serve as a roadmap for creating initial architectural design, as it conveniently highlights common decision points that need to be kept in mind when creating this kind of software. As the methodology is intended for the development of Web-based CAD/CAM software products to industrial standards, the methodology pays special attention to the quality of deliverables, by providing key considerations for ensuring application reliability, usability, robustness, performance, maintainability and security, as well as guidelines for meeting requirements for database capabilities, performance requirements, scalability and data integrity.

The methodology also provides advice on requirements elicitation, strategies for overcoming uncertainty during the project development and suggestions on resource allocation and collaboration.

9.1.4 Application of the methodology

The effectiveness of the Web-based CAD/CAM software development methodology has been examined using two case studies. The case studies were chosen based on the industrial focus of this research and incorporate the development of a Web-based CAD/CAM system for involute spur gear shaper cutters and a Web-based CNC code editor for online modification of the profile for manufacturing gear shaper cutters. Although the case studies are linked to gear cutting tool manufacturing, the methodology can be used for the development of CAD/CAM software in different context.

The development of these two industrial Web-based CAD/CAM applications allowed exploring in practice the challenges of developing industrial Web-based CAD/CAM systems and the effectiveness of the proposed methodology in overcoming the challenges. The methodology features, such as upstreaming design optimisation, exploratory development, priority based task planning coupled with project reserve management strategy, proved to be effective in the development of selected projects.

The upstreaming design optimisation enabled early estimation of advantages and risks associated with the implementation of different possible solutions, thus minimising the amount of unnecessary rework. The throwaway prototyping used for the evaluation of the system architecture at the beginning of the two case studies served well for the investigation of limitations governed by chosen techniques, algorithms and solutions and facilitated the elimination of important flaws in applications design. Using proof-of-concept prototyping and conducting exploratory development later in the project proved to be effective for reducing modelling uncertainty and timely elimination of the architectural drawbacks during the development of case studies. Prototyping also played an important role in requirements elicitation and initial assessment of the user interface design strategy.

The approach to planning based on time reserve management helped to cope with extended requirements and unforeseen problems identified late in the project development, contributing to the on-time delivery of the software product. The continuous monitoring and adjustments of the project reserve utilisation facilitated a good level of flexibility required for the exploratory development, while maintaining the main focus on the project objectives.

The composition of development team expertise and backgrounds and measures facilitating knowledge exchange and communication played an important role in the successful implementation of the case study projects and ensured high level of team performance throughout the development process.

9.1.5 Validation of the methodology

Effectiveness of the proposed methodology for Web-based CAD/CAM software development was checked through validation, evaluation and analysis of case study results. The validity of results produced by the Web-based CAD/CAM systems were validated using examples of real usage. Readiness of both systems for use in production was evaluated through industrial use of the software. Advantages of using the Web-based systems over similar desktop software were identified based on industrial feedback obtained with the use of questionnaires. Strengths and weaknesses of the developed Web-based CAD/CAM software were identified by analysing advantages and limitations of the developed software products.

The development of case studies using the established methodology resulted in on-time delivery of two industrial Web-based software, that produce valid results, embrace all business processes associated with the application area, ensure all functional and non-functional requirements and are used in production now. The developed software products demonstrate robustness, performance, reliability, security and usability comparable with the standards of modern commercial software and utilise advantages of Web-based applications to the highest extent.

Feedback from industry representatives indicates advantages of using the Web-based CAD/CAM systems over similar desktop applications, such as the capability of distributed access through the WWW, long-term implications of using Web-based technology due to the increased protection against changes in workstation software and hardware, capabilities for easy integration with other company operations, ease of use and training. The developed Web-based CAD/CAM system and CNC editor were given usability, interactivity and reliability estimates equally good or slightly better than those for desktop software, and unlike the desktop software the developed Web-based CAD/CAM systems received high collaboration and scalability estimates.

9.2 Discussion

The results of this research will have multiple effects, including direct impact on the GSCs manufacturing industry, influence on other areas of industry via delivering Web-based CAD/CAM solutions, as well as impact on the software development practices associated with the implementation of industrial CAD/CAM and CSE applications.

The research affects the GSCs manufacturing industry, firstly, by establishing a dedicated novel CAD/CAM approach to the modelling of the GSCs geometry, and secondly, by introducing Web technology to the area of CAD/CAM. The new CAD/CAM applications have improved competitiveness of the sponsor company by means of increasing the effectivity of designers' and operators' work, accumulating and re-using knowledge about the created products, as well as implementing improved design methods and techniques and increasing precision and quality of technical drawings and the end product. By using a novel analytical model, the system increases the flexibility in design and manufacture of gear shaper cutters, meeting the requirements of any gear producer. The browser-based implementation of the CAD/CAM software have radically changed related business processes by ensuring better accessibility, opportunities for user collaboration and customer involvement in product development process. The new software solutions will have long-term implications for the sponsor company, as they provide increased protection against changes in workstation software and hardware and offer capabilities for further improvement of business operations.

The methodology established in this research provides capabilities for facilitating the development of Web-based CAD/CAM solutions for other industries. Web-based applications have proven to be advantageous in many business areas and the new methodology will help to utilise these advantages in the area of CAD/CAM.

Finally, as the methodology for Web-based CAD/CAM development introduces some novel approaches for addressing well known software development challenges, the results of this research may be used by software development practitioners for improving the efficiency of software development projects having characteristics, listed in Section 5.9.

9.2.1 Major contributions

The overall contribution of this research is a methodology, which provides guidance on the development of Web-based CAD/CAM software to industrial quality standards in a time and cost effective manner. The methodology builds on features of widely used software models and proposes an optimised model for agile development of Web-based CAD/CAM systems and industrial science-intensive applications.

The methodology defines the scope of applicability, outlines major considerations and key principles to follow when developing this kind of software, describes an approach to requirements elicitation, resource allocation and collaboration, establishes strategies for overcoming uncertainty and describes the design concerns for industrial Web-based CAD/CAM systems. The crucial parts of the methodology are:

- A novel project development model facilitating architecture optimisation early in the project to minimise total development efforts, create future-proof solutions and ensure system maintainability;

- A novel approach for planning based on time reserve management and task prioritisation, which provides the flexibility required for exploratory development while maintaining the main focus on project objectives.

Chapter 6 and Chapter 7 describe the experience of applying the proposed methodology on real software development case studies, incorporating a Web-based CAD/CAM system for GSCs and a Web-based CNC code editor. Implementation of the case studies led to delivery of two industrial Web-based CAD/CAM applications, complying with business requirements and successfully deployed in production. The resulting software products ensure robustness, performance, reliability, security and usability comparable with the standards of modern commercial software. The two solutions do not have any proprietary components and are delivered as open source browser-based applications, utilising Web technology advantages to the highest extent.

In relation to the research gap identified in current literature, the main contribution of this research is a dedicated methodology providing a systematic approach to Web-based CAD/CAM software development, which is able to improve quality and reliability of Web-based CAD/CAM systems, enable their use in production and encourage creating more CAD/CAM systems using Web technology. The research also reviews design and development of industrial browser-based CAD/CAM systems, which appear examples of *built for Web* applications and utilise Web technology advantages to the highest extent. Finally, the research establishes considerable empirical evidence that demonstrates significant advantages of Web-based CAD/CAM software compared to similar desktop applications.

9.2.2 Limitations

The proposed methodology is largely unsuitable for small scope software projects with easily obtainable and well defined requirements, for predictable development processes that could be accurately scheduled using standard time and effort estimation techniques, or for projects that are difficult to decompose in smaller parts, which is necessary for the successful application of iterative development approach.

The limitations of this research are mainly associated with the validation of the proposed methodology. To test the suitability of the proposed methodology for the development of industrial Web-based CAD/CAM applications, validation has been performed through the application of the methodology in real projects and validation of resulting software. As the research project is limited in time and resources, only two case studies have been implemented for the validation of the proposed methodology.

It is difficult to compare the effectiveness of the proposed methodology with other methodologies, as in this case it would require developing similar software products using different methodologies side by side, while ensuring similar circumstances for the projects.

9.2.3 Future research

There could be three directions for further work in the continuation of this research: examination of the methodology on greater number of real projects, improving the description of the methodology and its comparison with other software development methodologies.

Application of the methodology for creating Web-based CAD/CAM systems on a number of real projects would allow demonstrating the repeatability of the results from this methodology and its robustness in dealing with the variations in project circumstances. Also utilising the proposed methodology for a large number of projects would confirm the identified prerequisites for its applicability and the specificity of the methodology for the development of Web-based CAD/CAM applications.

The description of the established methodology could be enhanced by establishing formal methods for project effort estimation and providing a list of existing software tools that could support software development process, such as project planning software, version control systems, automated testing and verification tools and others. In addition, dedicated software tools could be developed to support the specific approaches described in the methodology.

The description of the proposed methodology could provide better clarification on its applicability, effectiveness and technical limitations after applying it for a wider range of software development case studies. To address the technical limitations of the methodology for creating Web-based CAD/CAM systems, application of other software development methodologies, that are better suited for the stated conditions, is preferable.

For further research, evaluation and comparison of the proposed methodology with other software development approaches could be conducted using empirical and analytical means. The empirical approach assumes the development of similar CAD/CAM software projects using other methodologies along with the methodology introduced in this research and comparison of how successful are these projects. The analytical approach comprises formalisation of the components of the Web-based CAD/CAM software development methodology using definitions and descriptions of essence – the kernel and the language for software engineering methods [158], and further comparison with other software development methodologies.

9.3 Conclusions

The research demonstrates that complex Web-based science-intensive software could be developed for industrial applications in a time and cost effective manner. Also application of Web-based approach is proven to bring significant advantages in the area of CAD/CAM. The research describes a methodology for creating industrial Web-based CAD/CAM systems, which provides guidance on how to address the associated challenges.

The following objectives were accomplished for achieving the overall research aim:

1. *Provide the rationale and prerequisites for the new software development methodology.* Chapter 4 explains why the application of one of the existing methodologies

for the development of Web-based CAD/CAM software is troublesome and summarises the characteristics specific to these systems and related challenges in the development process. Then, the main prerequisites for the Web-based CAD/CAM software development methodology are formulated.

2. *Establish a methodology for Web-based CAD/CAM system development.* Chapter 5 provides key guidelines for Web-based CAD/CAM software development. It describes the scope of projects that could be successfully developed using the proposed methodology, and discusses the situations, that could cause difficulties in applying this methodology. Based on the challenges specific to Web-based CAD/CAM system development, identified in Chapter 4, major considerations are given to address associated issues and support the development process. Based on these considerations, key principles are outlined for developing Web-based CAD/CAM systems and project development process model is established. Special attention is paid to planning approach, strategies for overcoming uncertainty during project development and Web-based CAD/CAM software design concerns.

3. *Apply the proposed methodology to a set of CAD/CAM software development case studies.* In Chapter 6 and Chapter 7, the development of two Web-based CAD/CAM software applications using the proposed methodology is reviewed in detail. The chapter also incorporates a discussion on the encountered challenges and how the methodology helped to overcome them.

4. *While following the proposed methodology utilise Web technology advantages by developing industrial browser-based CAD/CAM software case studies.* Chapter 6 and Chapter 7 describe the development of two industrial CAD/CAM software case studies, implementing browser-based technologies, which are reviewed in Section 6.7 and Section 7.6.

5. *Perform validation of the proposed methodology by evaluating software developed using this methodology.* Validation of the proposed methodology is presented in Chapter 8 and incorporates examination of applications in production environment, detailed description of examples, industrial feedback and analysis of advantages and limitations of resulting software.

Comparison of the achievements made during the research against its aim concludes that the aim has been achieved, particularly the research presents:

- a novel methodology for creating Web-based CAD/CAM software systems;
- applications of the proposed methodology on real software development case studies, incorporating the Web-based CAD/CAM system for GSCs and the Web-based CNC code editor;
- examples of browser-based CAD/CAM system design and development;
- validation and evaluation of the proposed methodology;
- demonstration of Web-based CAD/CAM software advantages compared to similar desktop applications.

Conducting this research enabled to reveal how the Web-based CAD/CAM software is deferent from any other kind of software and how this could be addressed in a systematic way to enable its use in production and encourage creating more CAD/CAM systems using Web technology.

References

[1] M. Doig, Techtarget: What is software? - definition from whatis.com (November 1997), http://searchsoa.techtarget.com/definition/software, (accessed: 26 April 2012).

[2] D. Rose, Techtarget: What is application? - definition from whatis.com (August 2000), http://searchsoftwarequality.techtarget.com/definition/application, (accessed: 26 April 2012).

[3] Software system - eNotes.com Reference, http://www.enotes.com/topic/software_system, (accessed: 26 April 2012).

[4] BPC, Articles and Glossary: Application Development, http://www.bestpricecomputers.co.uk/glossary/application-development.htm, (accessed: 21 September 2011).

[5] New product development glossary, http://www.npd-solutions.com/glossary.html, (accessed: 21 September 2011).

[6] D. Green, A. DiCaterino, A survey of system development process models, Tech. rep., Center for Technology in Government, University at Albany / SUNY (February 1998).

[7] A. Abran, J. W. Moore, P. Bourque, R. Dupuis (Eds.), Guide to the Software Engineering Body of Knowledge (SWEBOK) 2004 Version, IEEE Computer Society, 2005.

[8] H. K. Klein, R. Hirschheim, Choosing between competing design ideals in information systems development, Information Systems Frontiers (2001) 75–90.

[9] W. Robson, Strategic Management and Information Systems: An Integrated Approach, 2nd Edition, Financial Times/ Prentice Hall, 1997.

[10] Selecting a development approach (2005), https://www.cms.gov/systemlifecycleframework/ downloads/selectingdevelopmentapproach.pdf, (accessed: 15 September 2011).

[11] S. Ellis, Frameworks, methodologies and processes (April 2008), http://www.vsellis.com/execution/frameworks-methodologies-and-processes/, (accessed: 25 April 2012).

[12] Definition of Web application in the Free Online Encyclopedia, http://encyclopedia2.thefreedictionary.com/web%20application.

[13] Web application, http://en.wikipedia.org/wiki/web_application, (accessed: 6/9/2011).

[14] McGraw-Hill's Encyclopedia of Science & Technology, 10th Edition, McGraw-Hill's AccessScience, The McGraw-Hill Companies.

[15] Harvard University Graduate School of Design - What is CAD/CAM?, http://www.gsd.harvard.edu/inside/cadcam/whatis.html, (accessed: 23 September 2008).

[16] Graduate education for computational science and engineering, http://www.siam.org/students/resources/report.php, (accessed: 6/9/2011).

[17] Computational Science and Engineering, ETH Zurich, http://www.cse.ethz.ch, (accessed: 8/9/2011).

[18] F. P. Brooks, The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley Professional, 1995.

[19] Dathan tool & gauge co. ltd., http://dathan.co.uk/, (accessed: 2 July 2014).

[20] A. Malahova, J. Butans, A. Tiwari, A web-based cad system for gear shaper cutters, in: Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on, 2009, pp. 37–42.

[21] Knowledge Transfer Partnerships (KTP) - Background, http://www.ktponline.org.uk/background/, (accessed: 24 July 2013).

[22] Web Server Survey - Netcraft, http://news.netcraft.com/archives/category/web-server-survey/, (accessed: 1 May 2012).

[23] T. Berners-Lee, R. Cailliau, J.-F. Groff, B. Pollermann, World-Wide Web: The Information Universe, Electronic Networking: Research, Applications and Policy 1 (2) (1992) 74–82.

[24] D. Brandon, Software Engineering for Modern Web Applications: Methodologies and Technologies, IGI Global, 2008.

[25] World Wide Web Tanth Anniversary - W3C Activities Timeline Graphic, http://www.w3.org/2004/talks/styles/w3c10/images/timeline.pdf, (accessed: 3 May 2012).

[26] Groups - W3C, http://www.w3.org/consortium/activities, (accessed: 3 May 2012).

[27] M. Jazayeri, Some trends in web application development, Faculty of Informatics, University of Lugano (USI), Lugano, Switzerland, 2007, pp. 199–213.

[28] Graphics - W3C, http://www.w3.org/standards/webdesign/graphics, (accessed: 1 May 2012).

[29] J. Toussaint, K. Cheng, Design agility and manufacturing responsiveness on the Web, Integrated Manufacturing Systems 13 (5) (2002) 328–339.

[30] D. Tony Liu, X. William Xu, A review of web-based product data management systems, Computers in Industry 44 (3) (2001) 251–262.

[31] Advanced manufacturing - technology: Interconnectivity the key to wired future, http://www.automationmag.com/ma-content/e-manufacturing/technology.html, (accessed: 11 November 2008).

[32] Gartner identifies the top 10 strategic technologies for 2008, http://www.gartner.com/it/page.jsp?id=530109.

[33] H. J. Wang, A. Moshchuk, A. Bush, Convergence of Desktop and Web Applications on a Multi-Service OS, in: 4th Usenix Workshop on Hot Topics in Security, Montreal, Canada, 2009.

[34] J. Zhao, C. Wang, R. Sun, B. Wang, A Citrix XenApp and web based system for product design and management, Applied Mechanics and Materials 163 (2012) 121–124.

[35] Benefits Of Web-Based Applications And Microsoft Announcement Of The ”Live” Era, http://www.masternewmedia.org/web-based_applications/web-based_applications_issues/benefits_of_web-based_applications_Microsot_announces_Live_era_20051102.htm, (accessed: 10.03.2008).

[36] F. Toni, M. A. Margevicius, M. A. Silver, Total cost of ownership comparison of pcs with server-based computing, Tech. rep., Gartner RAS Core Research Note G00159622 (August 2008).

[37] H. J. Kim, W. S. Chu, H. J. Kang, S. H. Ahn, D. S. Kim, C. S. Jun, Usability of web-based design and manufacturing systems: Case study of micro machining, ASME Conference Proceedings 2005 (47403) (2005) 921–927.

[38] X. Dai, Y. Qin, N. Juster, A web-based collaborative design advisory system for micro product design assessment, Applied Mechanics and Materials 10-12 (2008) 220–224.

[39] Y. Hu, R. Wang, Research on collaborative design software integration based on SOA, Journal of Advanced Manufacturing Systems 7 (1) (2008) 91 – 94.

[40] W. Tann, H.-J. Shaw, Enabling the web-based intelligent product design, in: Intelligent Systems Design and Applications, 2006. ISDA ’06. Sixth International Conference on, Vol. 3, 2006, pp. 14 –19.

[41] A. J. Alvares, J. C. E. Ferreira, R. M. Lorenzo, An integrated web-based CAD/CAPP/CAM system for the remote design and manufacture of feature-based cylindrical parts, Journal of Intelligent Manufacturing (2008) 1–17.

[42] MCAD Online - Dassault launches Catia V6 platform, http://www.mcadonline.com/index.php?option=com_content&task=view&id=513&itemid=73, (accessed: Monday, 25 February 2008).

[43] Website development and online marketing: Thomasnet web solutions: Thomas publishing, http://websolutions.thomasnet.com/, (accessed: 24 May 2012).

[44] P. Sypek, P. Kozakowski, M. Mrozowski, Internet as a new platform for delivery of microwave CAD services, Vol. 2, Tech. University of Gdansk, Department of Electronics, ul. Narutowicza 11/12, 80-952, Gdansk, 2004, pp. 574–577.

[45] L. Gordon, Let's get social., Machine Design 81 (10) (2009) 50 – 53.

[46] C. H. Chu, C. Y. Cheng, C. W. Wu, Applications of the web-based collaborative visualization in distributed product development, Computers in Industry 57 (3) (2006) 272–282.

[47] E. Ivask, A. Jutman, J. Raik, R. Ubar, Web-based framework for distributed remote laboratory in the field of digital system test, Tallinn, 2008, pp. 182–187.

[48] C. K. Mok, K. S. Chin, H. Lan, An internet-based intelligent design system for injection moulds, Robotics and Computer-Integrated Manufacturing 24 (1) (2008) 1–15.

[49] Y. Seo, D. Y. Kim, S. H. Suh, Development of Web-based CAM system, International Journal of Advanced Manufacturing Technology 28 (1-2) (2006) 101–108.

[50] J. Kang, B. Lho, S. Choi, Parametric web-cad for box culvert design, in: Computer Aided Design and Applications, Vol. 1, 2004, pp. 147–152.

[51] K. Zeng, Web-CAD - A CAD software based on the HTML5 canvas technology (2011), http://code.google.com/p/web-cad/.

[52] Z. Siddique, K. Boddu, A CAD template approach to support web-based customer centric product design, Journal of Computing and Information Science in Engineering 5 (4) (2005) 381–387.

[53] L. S. Indrusiak, A framework supporting collaboration on the distributed design of integrated systems, Ph.D. thesis, The Federal University of Rio Grande do Sul. Graduate Program in Computer Science (July 2003).

[54] L. Denis, Y. Gardan, E. Perrin, A framework for a distributed CAD system, CAD Computer Aided Design 36 (9) (2004) 761–773.

[55] S. F. Qin, R. Harrison, A. A. West, I. N. Jordanov, D. K. Wright, A framework of web-based conceptual design, Computers in Industry 50 (2) (2003) 153–164.

[56] M. M. Dessouky, S. Verma, D. E. Bailey, J. Rickel, A methodology for developing a web-based factory simulator for manufacturing education, IIE Transactions (Institute of Industrial Engineers) 33 (3) (2001) 167–180.

[57] J. Zeng, W. Chen, Q. Ding, A Web-based CAD system, Journal of Materials Processing Technology 139 (1-3 SPEC) (2003) 229–232.

[58] Alibre's progress, CAD/CAM Publishing, Inc. (September 2008).

[59] H. Y. Kan, V. G. Duffy, C. J. Su, An internet virtual reality collaborative environment for effective product design, Computers in Industry 45 (2) (2001) 197–213.

[60] J. S. Liang, P. W. Wei, Conceptual design system in a web-based virtual interactive environment for product development, International Journal of Advanced Manufacturing Technology 30 (11-12) (2006) 1010–1020.

[61] J. S. Liang, A web-based 3D virtual technologies for developing product information framework, International Journal of Advanced Manufacturing Technology 34 (5-6) (2007) 617–630.

[62] W. Liu, L. Zhou, H. Zhuang, C-superman: A web-based synchronous collaborative CAD/CAM system, Vol. 2, Res. Center of CAD/CAM Engineering, Nanjing Univ. Aero. and Astronaut., Nanjing, China, 2004, pp. 437–442.

[63] S. H. Ahn, V. Sundararajan, C. Smith, B. Kannan, R. D'Souza, G. Sun, A. Mohole, P. K. Wright, J. Kim, S. McMains, J. Smith, C. H. Sequin, CyberCut: An internet-based CAD/CAM system, ASME Journal of Computing and Information Science in Engineering 1 (1) (2001) 52–59.

[64] S. H. Ahn, B. Bharadwaj, H. Khalid, S. Y. Liou, P. K. Wright, Web-based design and manufacturing systems for automobile components: Architectures and usability studies, International Journal of Computer Integrated Manufacturing 15 (6) (2002) 555–563.

[65] J. Larson, H. H. Cheng, Object-oriented CAM design through the Internet, Journal of Intelligent Manufacturing 11 (6) (2000) 515–534.

[66] D. Fedasyuk, V. Makar, O. Markelov, System architecture of web-based CAD tool for thermal design of power semiconductor devices, CAD Department, Inst. Comp. Sci. and Info. Technol., Lviv Polytechnic National University, St. Bandery Str., 12, Lviv, 79013, Ukraine, 2004, pp. 286–288.

[67] C. Liu, Y. Zhang, L. Sun, Web based 3D assembly sequence planning prototype integrated with CAD model, in: Computer Supported Cooperative Work in Design, 2008. CSCWD 2008, IEEE, 2008, pp. 823 – 828.

[68] J. H. Wu, H. J. Yin, Web browser based CAD system for electrical machine, Zhongguo Dianji Gongcheng Xuebao/Proceedings of the Chinese Society of Electrical Engineering 27 (18) (2007) 35–40.

[69] P. Rajesh, B. Magesh-Babu, Web-based collaborative conceptual design: an XML approach, The International Journal of Advanced Manufacturing Technology 38 (5) (2008) 433–440.

[70] K. Cheng, P. Y. Pan, D. K. Harrison, Web-based design and manufacturing support systems: Implementation perspectives, International Journal of Computer Integrated Manufacturing 14 (1) (2001) 14–27.

[71] H. He, Y. Wu, Y. Gu, Y. Lu, Web-based virtual CNC machine modeling and operation, Chinese Journal of Mechanical Engineering (English Edition) 20 (6) (2007) 109–113.

[72] P. Jiang, X. Yan, Y. Liu, Service in e-design: A web "top-down" design tool for creating silicon micro-components, Journal of Manufacturing Technology Management 18 (1) (2007) 90–105.

[73] D. L. Z. O. Hamza-Lup, F.G.a, Web-based 3D planning tool for radiation therapy treatment, 2006, pp. 159–162.

[74] V. V. Satish, K. Motipalli, P. Krishnaswami, An innovative web application for design and automated NC code generation, Vol. 2006, Department of Mechanical and Nuclear Engineering, Kansas State University, Manhattan, KS 66502, 2006.

[75] G. Bolaños, S. Mann, S. Bedi, Targeted webcad, Computer-Aided Design and Applications 6 (5) (2009) 639–644.

[76] P. Guan, H. Zhang, T. Yu, W. Wang, Research on web-based virtual grinding machine tool, Advanced Materials Research 118-120 (2010) 825–829.

[77] K.-M. Cutler, CAD Users Rejoice: Sunglass Brings Slick 3D Modeling To The Browser. TechCrunch (May 22 2012), http://techcrunch.com/2012/05/22/cad-users-rejoice-sunglass-brings-slick-3d-modeling-to-the-browser/, (accessed: 29 June 2012).

[78] Web–based modeling and optimisation of logistics systems with W2MO (Web to Modeling and Optimization), http://www.logivations.com/en/intra-logistics/web-based-modeling-optimization.php.

[79] G. Q. Huang, K. L. Mak, Issues in the development and implementation of web applications for product design and manufacture, International Journal of Computer Integrated Manufacturing 14 (1) (2001) 125–135.

[80] R. Chaubey, J. K. Suresh, Integration vs. development: An engineering approach to building web applications, in: ASWEC '01: Proceedings of the 13th Australian Conference on Software Engineering, IEEE Computer Society, Washington, DC, USA, 2001, p. 171.

[81] 2012 trends report: Application security risks, Tech. rep., Cenzic, Inc. (March 11 2012).

[82] OWASP Guide Project (2010), https://www.owasp.org/index.php/owasp_guide_project.

[83] H. Cai, X. Tian, N. Li, B. Xu, Framework of geometry modeling integrated platform based on services for large-scale information visualization, Journal of Multimedia 5 (4) (2010) 361–368.

[84] E. Vezzetti, Product lifecycle data sharing and visualisation: Web-based approaches, The International Journal of Advanced Manufacturing Technology.

[85] G. Q. Huang, S. W. Lee, K. L. Mak, Synchronised web applications for product development in the 21st century, International Journal of Advanced Manufacturing Technology 18 (8) (2001) 605–613.

[86] H. Yang, D. Xue, Recent research on developing web-based manufacturing systems: A review, International Journal of Production Research 41 (15) (2003) 3601–3629.

[87] G. Q. Huang, K. L. Mak, Web-integrated manufacturing: Recent developments and emerging issues, International Journal of Computer Integrated Manufacturing 14 (1) (2001) 3–13.

[88] B. Boehm, A view of 20th and 21st century software engineering, in: Proceedings of the 28th international conference on Software engineering, ICSE '06, ACM, New York, NY, USA, 2006, pp. 12–29.

[89] P. Checkland, Soft systems methodology: a thirty year retrospective, Systems Research and Behavioral Science (17) (2000) 11–58.

[90] J. Mingers, J. Rosenhead, Rational analysis for a problematic world revisited: problem structuring methods for complexity, uncertainty and conflict, Wiley, 2001.

[91] C. Larman, V. R. Basili, Iterative and incremental development: A brief history, Computer 36 (2003) 47–56.

[92] C. Larman, Agile and Iterative Development: A Manager's Guide, Addison-Wesley Professional, 2003.

[93] M. Maier, Architecting principles for systems of systems, in: Proceedings of the Sixth Annual International Symposium, International Council on Systems Engineering, Boston, 1996, pp. 567–574.

[94] A. P. Sage, C. D. Cuppan, On the systems engineering and management of systems of systems and federations of systems, Inf. Knowl. Syst. Manag. 2 (4) (2001) 325–345.

[95] Forrester/Dr. Dobb's Global Developer Technographics Survey, Forrester Research, Inc. (Q3 2010).

[96] A. B. M. Moniruzzaman, S. A. Hossain, Comparative study on agile software development methodologies, ResearchGate.

[97] M. Bauer, Successful web development methodologies article, SitePoint (13 May 2012), http://www.sitepoint.com/successful-development/.

[98] CAD Schroer's Software Development Methodology, http://www.cad-schroer.com/services/development/1/1214/, (accessed: 1 October 2012).

[99] G. Sergia, A. Demurov, G. Petrosyan, R. Jobava, On One Approach to Scientific CAD/CAE Software Developing Process, International Journal of Computer Science Issues 7 (11) 9–15.

[100] M. Blom, Is Scrum and XP suitable for CSE Development?, Procedia Computer Science 1 (1) (2010) 1511 – 1517.

[101] F. Land, A contingency based approach to requirements elicitation and systems development, Journal of Systems and Software 40 (1) (1998) 3 – 6.

[102] F. Land, E. Somogyi, Software engineering: The relationship between a formal system and its environment, International Journal of Maritime Economics (1) (1986) 14–21.

[103] H. Naguib, Y. Li, (Position Paper) Applying software engineering methods and tools to CSE research projects, Procedia Computer Science 1 (1) (2010) 1505 – 1509.

[104] M. Heroux, J. Willenbring, Barely sufficient software engineering: 10 practices to improve your CSE software, Software Engineering for Computational Science and Engineering (2009) 15 – 21.

[105] M. Bajec, D. Vavpotic, M. Krisper, Practice-driven approach for creating project-specific software development methods, Information and Software Technology 49 (4) (2007) 345 – 365.

[106] M. G. Christel, K. C. Kan, Issues in requirements elicitation, Tech. rep., Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Pennsylvania 15213 (September 1992).

[107] J. Highsmith, Extreme programming, Tech. rep., Agile Project Management Advisory Service (2000).

[108] D. P. Shrivastava, R. V. Lakshminarayan, S. V. Sujatha, New engineering technique of software development, 2010, pp. 671–676.

[109] W. Stott, J. Newkirk, Improve the design and flexibility of your project with extreme programming techniques, MSDN Magazine.

[110] D. Sotirovski, Heuristics for iterative software development, IEEE Software 18 (3) (2001) 66–73.

[111] B. K. Jayaswal, P. C. Patton, Design for Trustworthy Software: Tools, Techniques, and Methodology of Developing Robust Software, 1st Edition, Prentice Hall, 2006.

[112] W. Wood, W. Kleb, Exploring XP for scientific research, IEEE Software 20 (3) (2003) 30–36+4.

[113] M. Fowler, The new methodology, http://www.martinfowler.com/articles/newmethodology.html, (accessed: 13 December 2005).

[114] R. A. Serway, Physics for Scientists and Engineers, 2nd Edition, Saunders College Publishing, 1986.

[115] E. M. Goldratt, Critical Chain. A Business Novel, North River Press, 1997.

[116] L. P. Leach, Critical chain project management, Artech House technology management and professional development library, Artech House, 2000.

[117] S. Balacco, C. Rommel, The increasing value and complexity of software call for the reevaluation of development and testing practices (April 2011).

[118] M. bin Mhd Fauzy, Finite element analysis of car alloy wheel, Tech. rep., Technical University of Malaysia Malacca. Faculty of Manufacturing Engineering (April 2008).

[119] S. H. Ahn, S. McMains, C. H. Séquin, P. K. Wright, Mechanical implementation services for rapid prototyping, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 216 (8) (2002) 1193–1199.

[120] M. Sarcar, K. Mallikarjunarao, K. Lalitnarayan, Computer Aided Design And Manufacturing, Prentice-Hall Of India Pvt. Limited, 2008.

[121] D. Straus, How to Make Collaboration Work (Large Print 16pt), Readhowyouwant, 2010.

[122] J. Metzler, A. Metzler, The 2011 Application and Service Delivery Handbook, Webtorials Editorial/Analyst Division, 2011.

[123] C. U. Smith, P. D, Best practices for software performance engineering, in: Proc. CMG, 2003.

[124] C. M. Hoffmann, Robustness in geometric computations, JCISE 1 (2001) 143–156.

[125] M. Nygard, Release It!: Design and Deploy Production-Ready Software, Pragmatic Bookshelf Series, , 2007.

[126] J. A. Gosden, Software compatibility: What was promised, what we have, what we need, in: Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I, AFIPS '68 (Fall, part I), ACM, New York, NY, USA, 1968, pp. 81–87.

[127] A. Sillitti, G. Succi, T. Vernazza, G.Granatella, T.Vernazza, Dealing with software compatibility, in: Proceedings of the 7th World Multi-Conference on Systemics, Cybernetics and Informatics, Orlando, FL, USA, 2003.

[128] Microsoft Application Architecture Guide, 2nd Edition, Microsoft Patterns & Practices, Microsoft Press, 2009.

[129] What makes software flexible, http://c2.com/cgi/wiki?whatmakessoftwareflexible, (accessed: 24 January 2013).

[130] Definition of extendible at askdefine, online english dictionary, http://extendible.askdefine.com/, (accessed: 25 January 2013).

[131] M. Aimonetti, Designing for scalability, http://merbist.com/2011/01/31/designing-for-scalability/, (accessed: 31 January 2011).

[132] S. Brown, Scalability principles, http://www.infoq.com/articles/scalability-principles, (accessed: 21 May 2008).

[133] I. Sommerville, Software Engineering 7, Icss Series, New York : Pearson/Addison-Wesley, 2004.

[134] P. R.Manjula, E. A. Sriram, Article:reliability evaluation of web applications from click-stream data, International Journal of Computer Applications 9 (5) (2010) 23–29.

[135] G. McGraw, Software Security: Building Security in, Addison-Wesley professional computing series, Addison-Wesley, 2006.

[136] B. Shneiderman, Designing the user interface: strategies for effective human-computer-interaction, No. v. 85 in Designing the User Interface: Strategies for Effective Human-computer-interaction, Addison Wesley Longman, 1998.

[137] C. Jones, O. Bonsignour, J. Subramanyam, The Economics of Software Quality, Addison-Wesley, 2011.

[138] M. Stamatelatos, Probabilistic Risk Assessment: What Is It And Why Is It Worth Performing It?, NASA Office of Safety and Mission Assurance.

[139] J. Davis, A. Fusfeld, E. Scriven, G. Tritle, Determining a project's probability of success, Research-Technology Management 44 (3) (2001) 51–57(7).

[140] P. E. A. Lewis M. Branscomb, Taking Technical Risks: How Innovators, Managers, and Investors Manage Risk in High-Tech Innovations, MIT Press, 2003.

[141] W. D. Rowe, Managing uncertainty, Risk Analysis (14) (1994) 743–750.

[142] A. DeMeyer, C. H. Loch, M. T. Pich, A framework for project management under uncertainty, Sloan Management Review 43 (2) (2002) 60–67.

[143] M. T. Pich, C. H. Loch, A. D. Meyer, On uncertainty, ambiguity, and complexity in project management, Manage. Sci. 48 (2002) 1008–1023.

[144] Stakeholder commitment (September 2006), http://www.spspro.com/brochures/6-stakeholder(accessed: 12 February 2013).

[145] C. Beringer, D. Jonas, A. Kock, Behavior of internal stakeholders in project portfolio management and its impact on success, International Journal of Project Management (0) (2013) –.

[146] E. Warren, Collaboration and productivity in the workplace (September 2011), http://engagementstrategies.com/collaboration-and-productivity-in-the-workplace/, (accessed: 14 February 2013).

[147] D. N. Warner, D. M. Letsky, D. M. Cowen, Structural model of team collaboration (2003).

[148] J. Butans, Computer aided design of gear cutting tools, Master's thesis, School of Applied Sciences, Cranfield University (2008).

[149] E. Buckingham, Analytical mechanics of gears, McGraw-Hill Book Co, 1949.

[150] F. L. Litvin, Theory of gearing, Tech. rep., AVSCOM, available in Cranfield Library (1989).

[151] F. Di Puccio, M. Gabiccini, M. Guiggiani, Alternative formulation of the theory of gearing, Mechanism and Machine Theory 40 (2005) 613–637.

[152] DoD News Briefing - Secretary Rumsfeld and Gen. Myers (February 12 2002), http://www.defense.gov/transcripts/transcript.aspx?transcriptid=2636.

[153] M. Jørgensen, A review of studies on expert estimation of software development effort, Journal of Systems and Software 70 (1-2) (2004) 37–60.

[154] M. Jørgensen, B. Boehm, Software development effort estimation: Formal models or expert judgment?, http://simula.no/research/se/publications/simula.se.321/simula_pdf_file.

[155] M. Jorgensen, Practical guidelines for expert-judgment-based software effort estimation, IEEE Software 22 (3).

[156] M. Molokken, K.; Jorgensen, A review of software surveys on software effort estimation, Empirical Software Engineering (2003) 223 – 230.

[157] G. Booch, The architecture of web applications, http://www.ibm.com/developerworks/ibm/library/it-booch_web/, (accessed: 30 September 2009).

[158] Object Management Group, Inc., Essence – Kernel and Language for Software Engineering Methods, essence, revised submission Edition (November 2012).

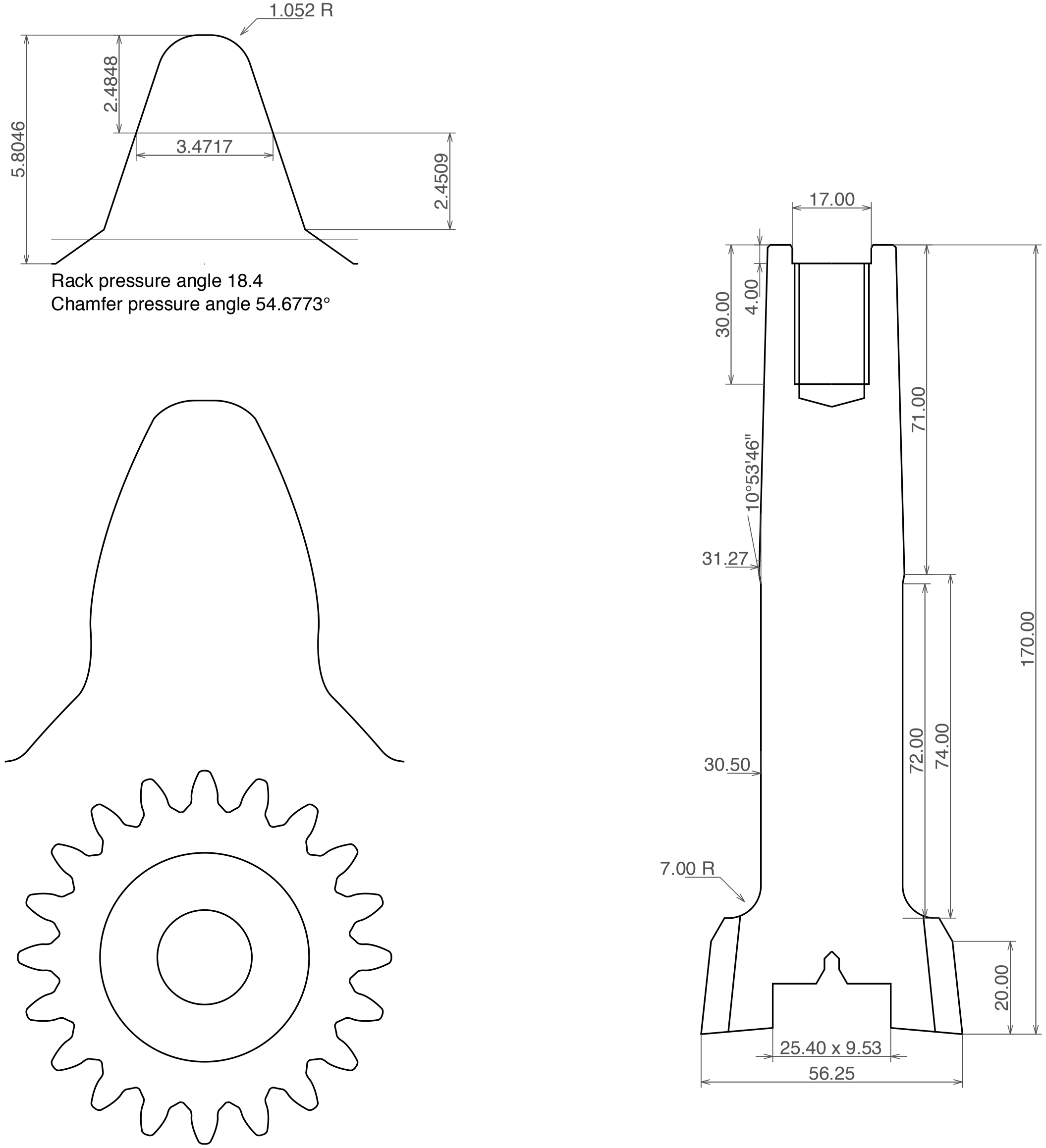
Appendix A

Design case 1

A.1 Web-based CAD/CAM system outputs

A.1.1 Tool drawing

BASIC RACK OF CUTTER TOOTH IN CUTTING PLANE



ALL DIMENSIONS IN MILLIMETRES

TOOL DESIGN DETAILS			
PRESSURE ANGLE	18.4000	MODULE	2.4758
NUMBER OF TEETH	20	PITCH CIRCLE DIAMETER	49.5161
BASE CIRCLE DIAMETER	46.9402	CIRCULAR TOOTH TH. AT NOMINAL DIA.	3.4717
MINIMUM DEPTH	5.6012	CUTTING DEPTH	5.1946
CIRCULAR PITCH	7.7780	SIDE RELIEF ANGLE	2.0621
PROTUBERANCE	NONE	TIP RADIUS	0.5000
NOMINAL OUTSIDE DIAMETER	54.4857	SPAN OVER 3 TEETH AT NOM. DIA.	18.5959
TOP RAKE	6.1200	FRONT RAKE	5.0000
NO.4 MORSE TAPER		16mm SI THREAD	
EXTERNAL PART DETAILS			
MAJOR DIAMETER	253.0000	MINOR DIAMETER MAX MIN	242.6380 242.5380
MODULE	2.5000	NUMBER OF TEETH	100
CIRCULAR THICKNESS	3.5011	PRESSURE ANGLE	20.0000
T.I.F. DIAMETER		MEASUREMENT OVER D=4.0000 PINS	253.8080 253.6610
DESCRIPTION OF THE TOOL GRADE A SHANK TYPE SPUR GEAR SHAPER CUTTER 2.5 MOD 20 P.A.			
DATHAN TOOL & GAUGE CO. LTD		Mean Lane, Meltham, Holmfirth, West Yorkshire, HD9 5RU, England	
DATHAN REFERENCE	812 / 116 / A / 1		
COMPONENT NO 32413540232			
CUSTOMER TOOL NO AA01004522			
CUSTOMER ZF PADOVA (FUBRI)			
ROCKWELL C	66/67	MAT'L	ASP 2030 HSS
DESIGNED BY	ANNA MALAHOVA	DATE	14 MAR 2013

A.1.2 Design details

Design Details

All dimensions in millimetres

Inputs

External master gear details

Diametral pitch	10.16000	Pressure angle	20.00000
No of teeth	100	Circular tooth thickness	3.50106
Major diameter max	253.00000	Minor diameter max	242.63800
Major diameter min	-	Minor diameter min	242.53800
Max measurement over D=4.0000 pins	253.8080	Min measurement over D=4.0000 pins	253.6610
Chamfer diameter max	252.60000	Chamfer angle	55.00000
Chamfer diameter min	252.50000		

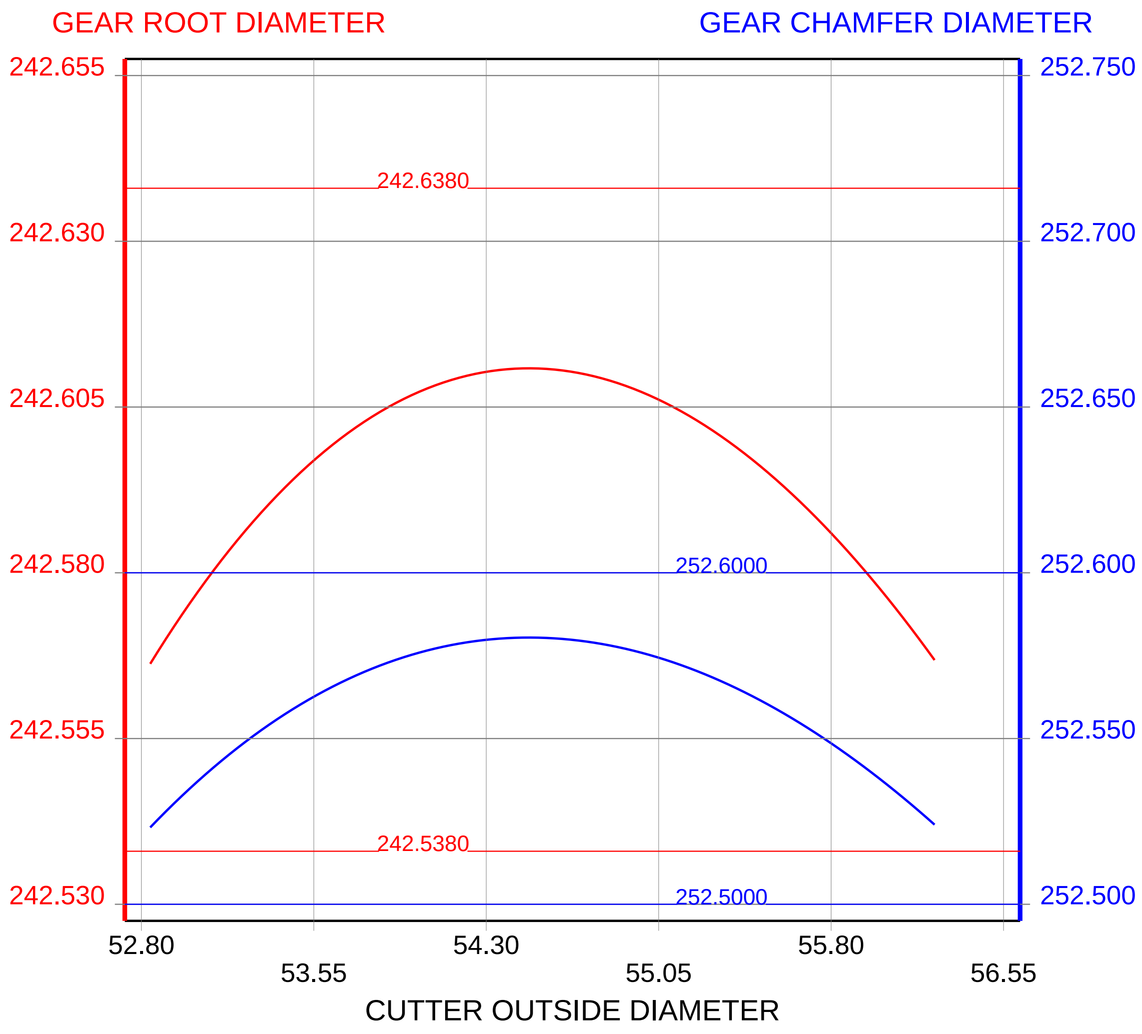
Cutter details

Diametral pitch	10.25929	Design pressure angle	18.40000
No of teeth	20	Circular tooth thickness	3.47168
Front rake	5.00000	Top rake	6.12000
Oversize	1.76430	Clearance	0.61000
Life for graphs	16.00000	Height of fillet	0.72000
Rack root radius	-		
Manual root diameter	242.61084	Manual chamfer diameter	252.58046

Results

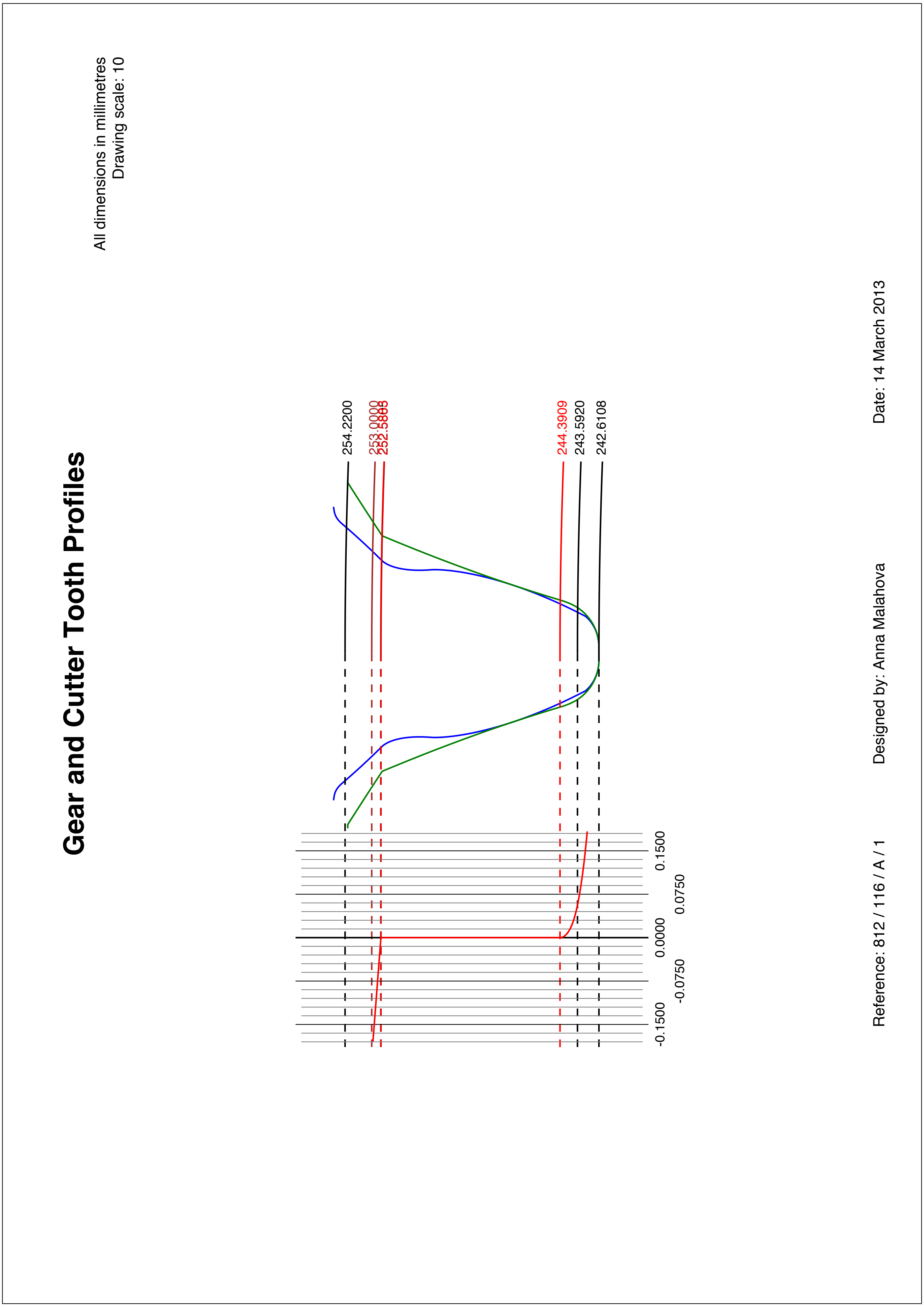
Addendum	2.48481	Dedendum	2.70977
Finished circular tooth thickness	4.05859	Pressure angle at finished OD	33.35476
Former tip thickness	2.09760	Tip thickness at finished OD	0.97787
Nominal outside diameter	54.48571	Finished outside diameter	56.25001
Former pressure angle	18.66140	Corrected pressure angle	18.56235
Correction factor	0.98497	Side relief angle	2.06206
Radius of fillet	1.05209	Height of fillet	0.72000
Max height of fillet	0.86277	Min height of fillet	0.59737
Base circle diameter	46.94015	Pitch circle diameter	49.51609

Reference: 812 / 116 / A / 1 Designed by: Anna Malahova Date: 14 March 2013



Reference: 812 / 116 / A / 1 Designed by: Anna Malahova Date: 14 March 2013

A.1.3 Multigraph


Gear and Cutter Tooth Profiles
All dimensions in millimetres
Drawing scale: 10
254.2200
244.3909
243.5920
242.6108
0.1500
0.0750
0.0000
-0.0750
-0.1500
Reference: 812 / 116 / A / 1
Designed by: Anna Malahova
Date: 14 March 2013

A.1.4 Job ticket

REF 812 / 116 / A **20 T**

Pitch	10.2593	PA	18.4	
P.C.D.	1.9495	Min depth	0.2205	
OD Theo	2.1451	OD Fin	2.2146	
Theo ht Theo th	0.1002 0.1366	Theo span T spanned	0.7321 3	
Fin ht Fin th	0.1358 0.1596	Fin span T spanned	0.7540 3	
Front rake	5°	Top rake	6°7'12"	
Cut OD		Cut depth	0.2251	
Rotary		Shaper	254/18	A
Divide		Former	812/116	
Fitting	No.4 Morse	Fitting flat		
Recess dia depth	1.0000 0.3750	Thread	16mm SI	
External gear		Front ctr	Large	
Base dia	1.8480	Circ. th.	0.1367	
OD max	2.2246	OD min	2.2046	
Grade	A	Tip relief	None	
Tip rad	0.0197	Prot.	None	
S.R.A.	2.0621	Scale	1	

ZF PADOVA (FUBRI)

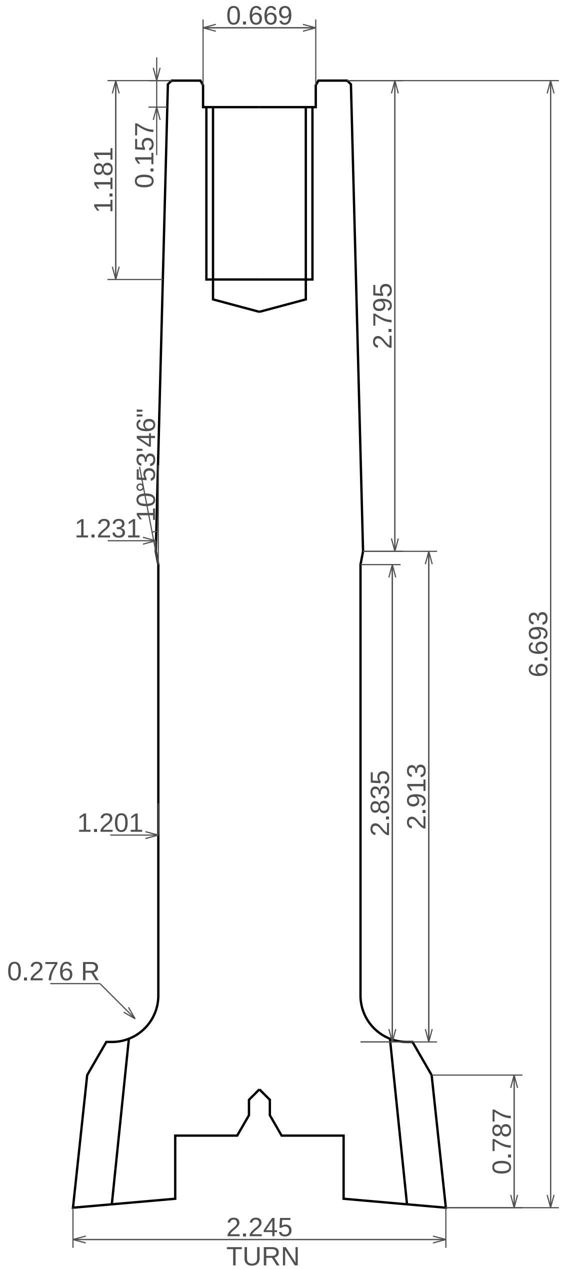
ETCH

```
Mn 2.5              APn 20            Z 100

H 5.195 mm                      REF 812/116

.             AA01004522

.
```

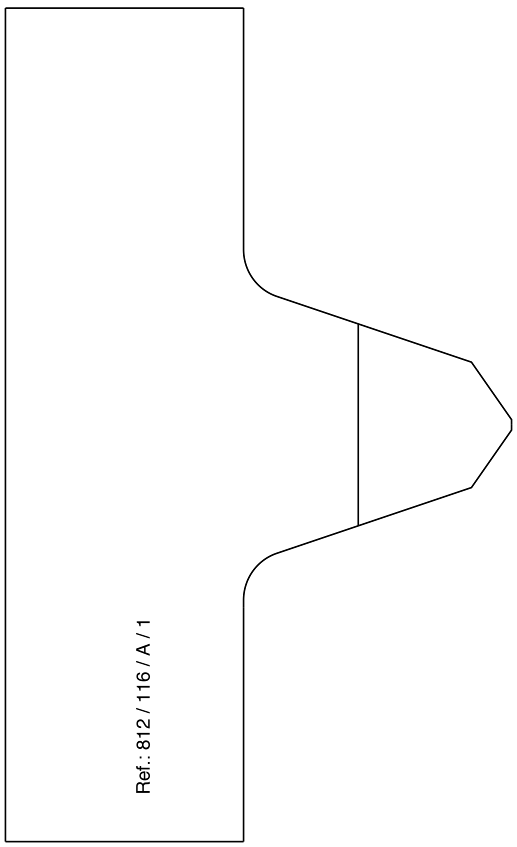


Date: 14 March 2013

Designed by: Anna Malahova

A.1.5 Former

HEIGHT	1.2874
THICKNESS	1.6954
ADDENDUM	0.9636
FORMER P.A.	18°39'41"
TOTAL DEPTH	2.2509
RADIUS OF FILLET	0.4106
HEIGHT OF FILLET	0.2792
FORMER SCALE	10.0
PLATE WIDTH	7.0
TOOL DIAMETER	0.2500
REFERENCE	812 / 116 / A / 1
CHAMFER P.A.	55°5'8"
HEIGHT TO CHAMFER	1.9140



A.1.6 Haas CNC instructions

```
%
O812
(812/116)
(T1//.25)
(T2//.1875)
(T3//.125)
(T4//.0625)
N10T1M06
(=N310/2)
G43D1H1
M97P1000L2
M30
N1000
N20G0G20G90G54X4.0000Y-0.5000
N30G0Z-1.0000
N40S3600M3
M8
N50G1F10.00
N60G41X4.0000Y-0.5000
N70X4.0000Y0.0000
N75X1.4668Y-0.0000
N80X1.4571Y-0.0003
N85X1.4478Y-0.0007
N90X1.4389Y-0.0013
N95X1.4303Y-0.0021
N100X1.4221Y-0.0030
N105X1.4064Y-0.0051
N110X1.3917Y-0.0077
N115X1.3778Y-0.0107
N120X1.3646Y-0.0139
N125X1.3521Y-0.0174
N130X1.3343Y-0.0232
N135X1.3230Y-0.0273
N140X1.3069Y-0.0337
N145X1.2918Y-0.0406
N150X1.2774Y-0.0478
N155X1.2637Y-0.0552
N160X1.2508Y-0.0630
N165X1.2384Y-0.0710
N170X1.2305Y-0.0764
N175X1.2228Y-0.0820
N180X1.2118Y-0.0904
N185X1.2047Y-0.0962
N190X1.1978Y-0.1020
N195X1.1879Y-0.1110
N200X1.1815Y-0.1170
N205X1.1754Y-0.1231
N210X1.1665Y-0.1324
N215X1.1607Y-0.1386
N220X1.1552Y-0.1450
N225X1.1498Y-0.1514
N230X1.1447Y-0.1578
N235X1.1396Y-0.1643
N240X1.1348Y-0.1709
N245X1.1278Y-0.1808
N250X1.1190Y-0.1943
N255X1.1108Y-0.2080
N260X1.1033Y-0.2218
N265X1.0963Y-0.2359
N270X1.0899Y-0.2501
N275X1.0841Y-0.2646
N280X1.0788Y-0.2792
N285X0.5267Y-1.9140
N290X0.0440Y-2.2509
N295X-0.0440Y-2.2509
N300X-0.5267Y-1.9140
N305X-1.0788Y-0.2792
N310X-1.0841Y-0.2646
N315X-1.0899Y-0.2501
N320X-1.0963Y-0.2359
```

```
N325X-1.1033Y-0.2218
N330X-1.1108Y-0.2080
N335X-1.1190Y-0.1943
N340X-1.1278Y-0.1808
N345X-1.1348Y-0.1709
N350X-1.1396Y-0.1643
N355X-1.1447Y-0.1578
N360X-1.1498Y-0.1514
N365X-1.1552Y-0.1450
N370X-1.1607Y-0.1386
N375X-1.1665Y-0.1324
N380X-1.1754Y-0.1231
N385X-1.1815Y-0.1170
N390X-1.1879Y-0.1110
N395X-1.1978Y-0.1020
N400X-1.2047Y-0.0962
N405X-1.2118Y-0.0904
N410X-1.2228Y-0.0820
N415X-1.2305Y-0.0764
N420X-1.2384Y-0.0710
N425X-1.2508Y-0.0630
N430X-1.2637Y-0.0552
N435X-1.2774Y-0.0478
N440X-1.2918Y-0.0406
N445X-1.3069Y-0.0337
N450X-1.3230Y-0.0273
N455X-1.3343Y-0.0232
N460X-1.3521Y-0.0174
N465X-1.3646Y-0.0139
N470X-1.3778Y-0.0107
N475X-1.3917Y-0.0077
N480X-1.4064Y-0.0051
N485X-1.4221Y-0.0030
N490X-1.4303Y-0.0021
N495X-1.4389Y-0.0013
N500X-1.4478Y-0.0007
N505X-1.4571Y-0.0003
N510X-1.4668Y-0.0000
N520X-4.0000Y0.0000
N530G40X-4.0000Y-0.5000
N540G0Z0
G43T21D21H21
M99
%
```

Appendix B

Design case 2

B.1 Design data

Table B.1: Cutter details

Parameter	Value
Standard	Custom
Diametral pitch	3 mm
Pressure angle	30°
No of teeth	12
Addendum	3.80833 mm
Dedendum	2.11667 mm
Circular tooth thickness	17.69682 mm
Fillet type	Round
Height of fillet	0.8382 mm
Oversize	0 mm
Clearance	2.5 mm
Top rake	4.25°
Front rake	5°
Life for graphics	14 mm

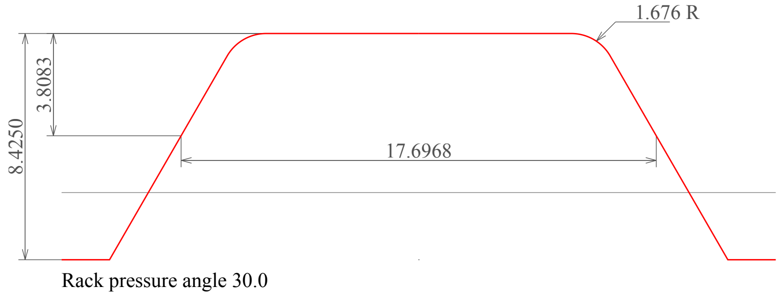


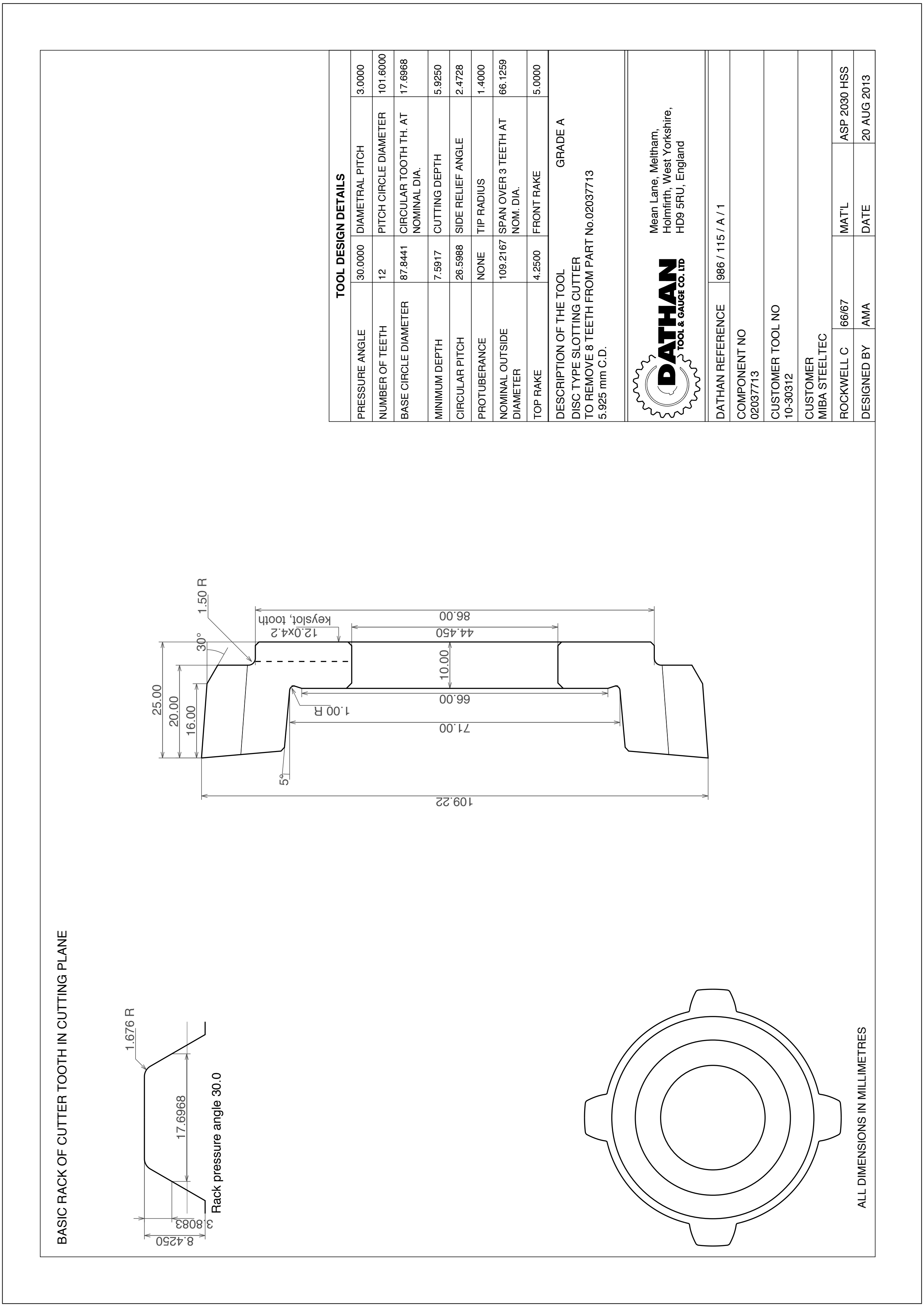
Figure B.1: Rack tooth profile

Table B.2: Tool body details

Parameter	**Value**
Body shape	Disk
Bore size	44.450 mm
Body type	DIN1825_100
Back boss type	Back boss I
Thickness	25
Tooth length	16 mm
Back land	5 mm
Web thickness	10 mm
Outer recess diameter	71
Inner recess diameter	66 mm
Back boss diameter	86 mm
Recess angle	5°
Chamfer angle	30°
Back boss angle	0°
Recess radius	1 mm
Back radius	1.5 mm
Keyslot / Keyway	
Key	Slot
Aligned on	Tooth
Width	12
Depth	4.2

B.2 Web-based CAD/CAM system outputs for design case 2

B.2.1 Tool drawing


TOOL DESIGN DETAILS
PRESSURE ANGLE
30.0000
DIAMETRAL PITCH
3.0000
NUMBER OF TEETH
12
PITCH CIRCLE DIAMETER
101.6000
BASE CIRCLE DIAMETER
87.8441
CIRCULAR TOOTH TH. AT NOMINAL DIA.
17.6968
MINIMUM DEPTH
7.5917
CUTTING DEPTH
5.9250
CIRCULAR PITCH
26.5988
SIDE RELIEF ANGLE
2.4728
PROTUBERANCE
NONE
TIP RADIUS
1.4000
NOMINAL OUTSIDE DIAMETER
109.2167
SPAN OVER 3 TEETH AT NOM. DIA.
66.1259
TOP RAKE
4.2500
FRONT RAKE
5.0000
DESCRIPTION OF THE TOOL
GRADE A
DISC TYPE SLOTTING CUTTER
TO REMOVE 8 TEETH FROM PART No.02037713
5.925 mm C.D.
DATHAN
TOOL & GAUGE CO. LTD
Mean Lane, Meltham,
Holmfirth, West Yorkshire,
HD9 5RU, England
DATHAN REFERENCE
986 / 115 / A / 1
COMPONENT NO
02037713
CUSTOMER TOOL NO
10-30312
CUSTOMER
MIBA STEELTEC
ROCKWELL C
66/67
MAT'L
ASP 2030 HSS
DESIGNED BY
AMA
DATE
20 AUG 2013
1.50 R
keyslot, tooth
12.0x4.2
86.00
44.450
30°
10.00
25.00
20.00
16.00
66.00
1.00 R
71.00
5°
109.22
BASIC RACK OF CUTTER TOOTH IN CUTTING PLANE
1.676 R
17.6968
Rack pressure angle 30.0
3.8083
8.4250
ALL DIMENSIONS IN MILLIMETRES

B.2.2 Design details

Design Details

All dimensions in millimetres

Inputs

Cutter details

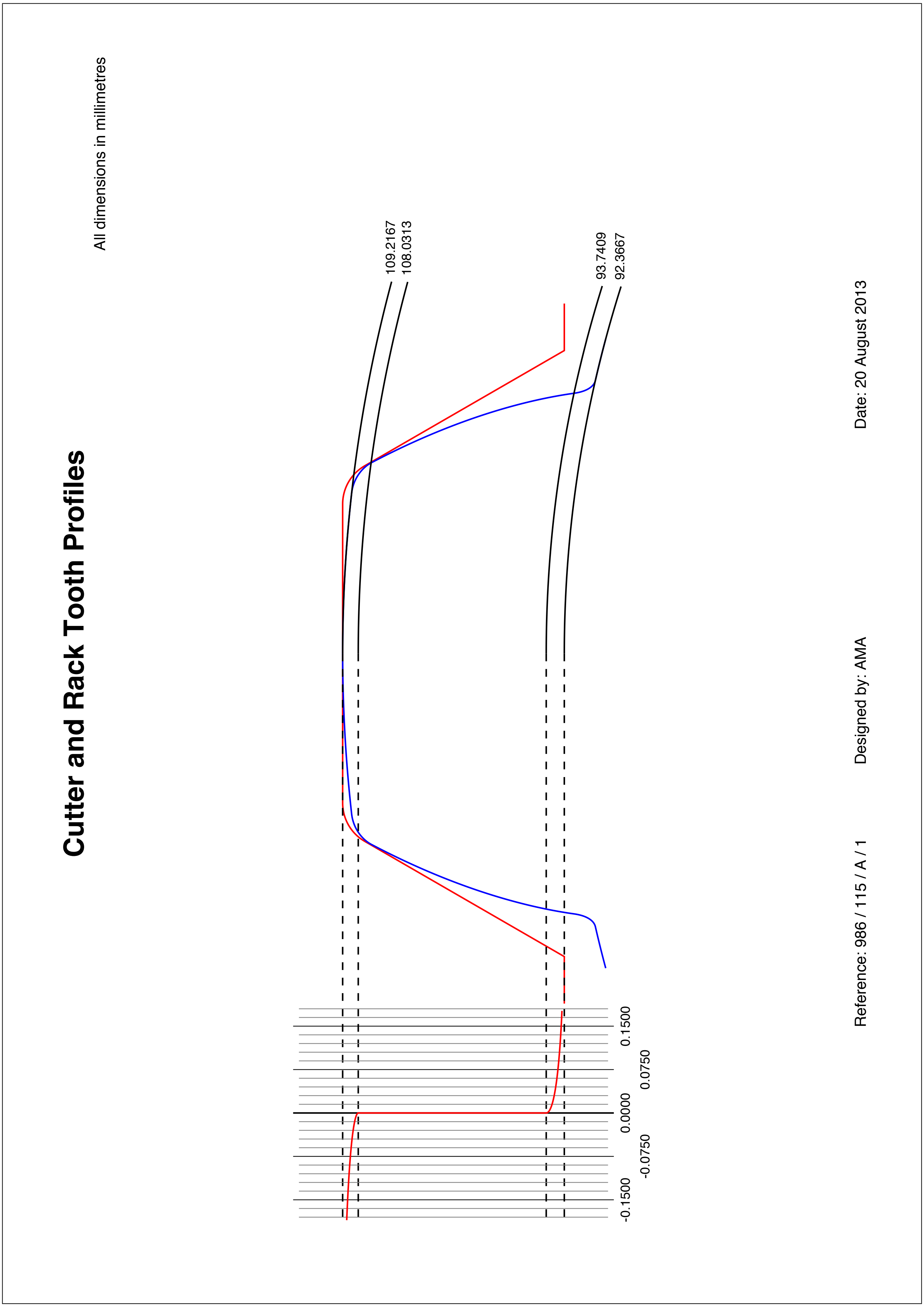
Diametral pitch	3.00000	Design pressure angle	30.00000
No of teeth	12	Circular tooth thickness	17.69682
Front rake	5.00000	Top rake	4.25000
Oversize	0.00000	Clearance	2.50000
Life for graphs	14.00000	Height of fillet	0.83820
Rack root radius	-		

Results

Addendum	3.80833	Dedendum	2.11667
Finished circular tooth thickness	17.69682	Pressure angle at finished OD	36.32889
Former tip thickness	3.57113	Tip thickness at finished OD	13.83141
Nominal outside diameter	109.21666	Finished outside diameter	109.21666
Former pressure angle	30.23068	Corrected pressure angle	30.16209
Correction factor	0.99077	Side relief angle	2.47278
Radius of fillet	1.67640	Height of fillet	0.83820
Max height of fillet	5.75878	Min height of fillet	0.33281
Base circle diameter	87.84411	Pitch circle diameter	101.60000

Reference: 986 / 115 / A / 1 Designed by: AMA Date: 20 August 2013

B.2.3 Multigraph

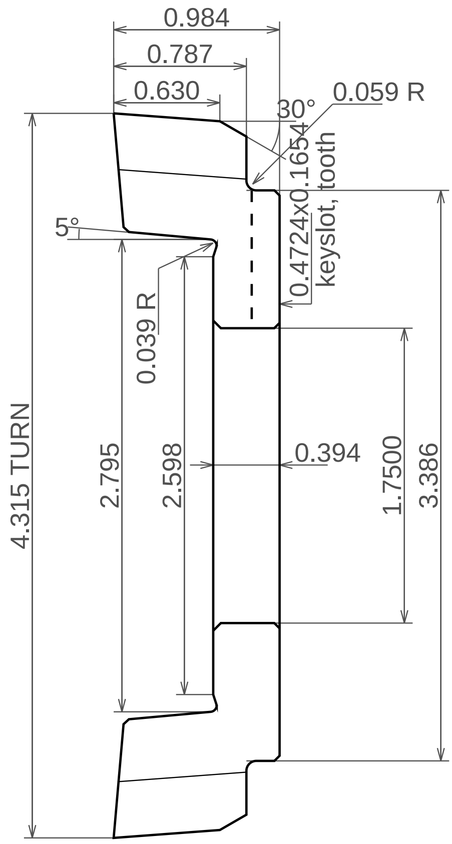

Cutter and Rack Tooth Profiles
All dimensions in millimetres
109.2167
108.0313
93.7409
92.3667
0.1500
0.0750
0.0000
-0.0750
-0.1500
Reference: 986 / 115 / A / 1
Designed by: AMA
Date: 20 August 2013

B.2.4 Job ticket

REF 986 / 115 / A 12 T 1.7500 Bore

Pitch	3.0000	PA	30.0
P.C.D.	4.0000	Min depth	0.2989
OD Theo	4.2999	OD Fin	4.2999
Theo ht Theo th	0.1802 0.6932	Theo span T spanned	2.6034 3
Fin ht Fin th	0.1802 0.6932	Fin span T spanned	2.6034 3
Front rake	5°	Top rake	4°15'
Cut OD			
Rotary			
Divide		Former	986/115
Base dia	3.4584	Circ. th.	0.6967
OD max	4.3099	OD min	4.2899
Grade	A	Tip relief	None
Tip rad	0.0551	Prot.	None
S.R.A.	2.4728	Scale	1

MINA STEELTEC 10-30312



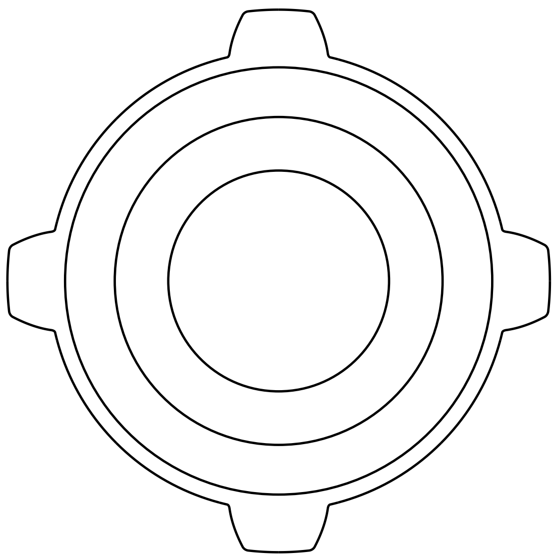
ETCH

SLOTTING CUTTER TO REMOVE 8 TEETH
FROM PART No. 02037713

5.925 mm REF 986/115

. TOOL No. 10-30312

.



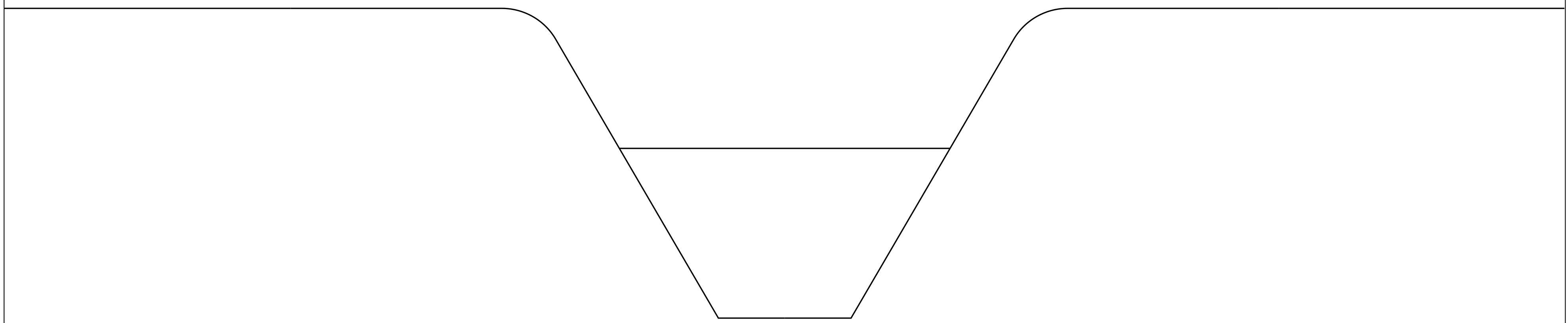
SCALE 0.75

Date: 20 August 2013

Designed by: AMA

B.2.5 Former

Ref.: 986 / 115 / A / 1



HEIGHT	1.8008
THICKNESS	3.5047
ADDENDUM	1.4855
FORMER P.A.	30°13'50"
TOTAL DEPTH	3.2863
RADIUS OF FILLET	0.6585
HEIGHT OF FILLET	0.3270
FORMER SCALE	10.0
PLATE WIDTH	16.8
TOOL DIAMETER	0.2500
REFERENCE	986 / 115 / A / 1

B.2.6 Haas CNC instructions

```
%
O986
(986/115)
(T1//.25)
(T2//.1875)
(T3//.125)
(T4//.0625)
N10T1M06
G43D1H1
M97P1000L2
M30
N1000
N20G0G20G90G54X8.9000Y-0.5000
N30G0Z-1.0000
N40S3600M3
M8
N50G1F10.00
N60G41X8.9000Y-0.5000
N70X8.9000Y0.0000
N80G1X5.2360Y0.0000
N85G1X2.9991Y0.0000
N90G3X2.6671Y-0.0887R0.6653
N95G3X2.5310Y-0.1929R0.6625
N100G3X2.4275Y-0.3270R0.6502
N105G1X0.7030Y-3.2863
N110G1X-0.7030Y-3.2863
N115G1X-2.4275Y-0.3270
N120G3X-2.5310Y-0.1929R0.6502
N125G3X-2.6671Y-0.0887R0.6625
N130G3X-2.9991Y-0.0000R0.6653
N135G1X-5.2360Y0.0000
N145G1X-8.9000Y0.0000
N155G40X-8.9000Y-0.5000
N165G0Z0
G43T21D21H21
M99
%
```

Appendix C

Design case 3

C.1 Design data

Table C.1: Internal master gear details

Parameter	Value
Gear type	Internal
MOD	1.6
Pressure angle	30°
No of teeth	9
Major diameter max	17.18 mm
Major diameter min	17 mm
Minor diameter min	14 mm
Circular tooth thickness	1.6818 mm
Required T.I.F.	16.7 mm
Chamfer parameters	
Chamfer diameter max	14.6 mm
Chamfer diameter min	14.55 mm
Chamfer angle	60°
Redesign	
Redesign type	Simple
Pressure angle	37.5°
Manual root diameter	17.1 mm

Figure C.1 illustrates trimming prediction. The check for trimming on internal gear is performed using sharp corner calculations and round corner calculations. The sharp corner calculations assume that gear and cutter have sharp corners on their teeth, which is the worst case approximate answer. These calculations are simple and fast, and are performed every time the design data are calculated. Both tooth and space feeds are checked automatically and the largest interference is reported to the user. However,

Table C.2: Cutter details

Parameter	Value
No of teeth	6
Fillet type	Round
Height of fillet	1 mm
Oversize	0 mm
Clearance	0.25 mm
Top rake	3°
Front rake	5°
Life for graphics	6 mm

sharp corner calculations cannot handle the cases wherein the gear teeth extend past the base diameter. In turn, round corner calculations perform profile interference analysis using the exact shapes of the gear and cutter teeth. This allows any tooth geometry to be handled with high accuracy, however, these calculations require more time to run and are used only when user works with trimming prediction module. For the given design case the results of sharp and round corner trimming calculations are different. The sharp corner trimming calculations indicate minimum cutter tooth feed clearance equal to approximately 0.1570 mm at tooth 2, and the round corner trimming calculations display minimum cutter tooth feed clearance at tooth 2 equal to approximately 0.3146 mm.

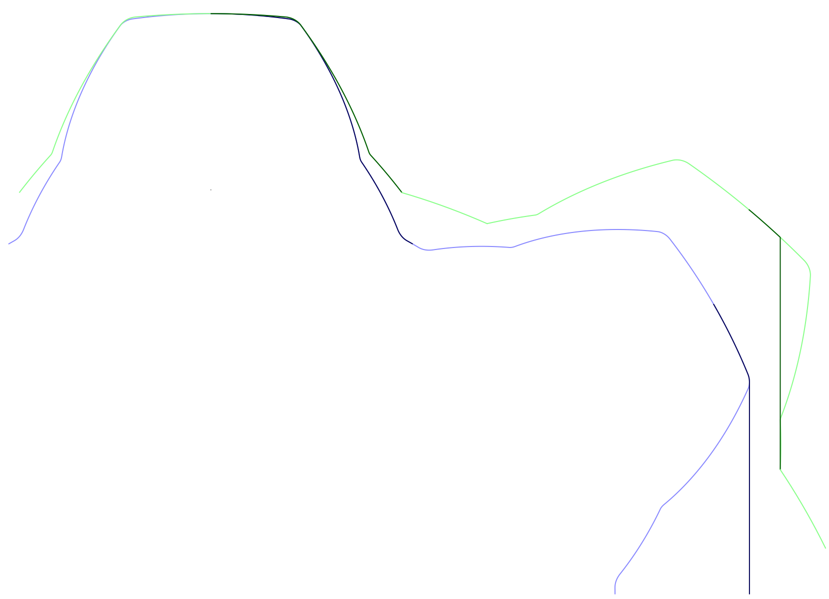


Figure C.1: Cutter (blue) and internal gear (green) teeth profile interference analysis. The distance between the dark vertical lines is the minimum cutter tooth feed clearance.

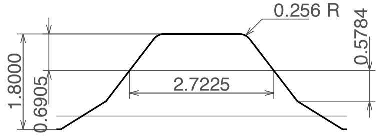
Table C.3: Tool body details

Parameter	Value
Body shape	Shank
Tooth length	8 mm
Fitting type	No. 2 Morse
Thread type	6 mm SI
Neck diameter	10 mm
Neck length	22 mm
Ramp radius	4 mm
Rib length	17 mm
Rib radius	3 mm
Fitting length	40 mm
Fitting top diameter	18 mm
Taper angle	1.4307 mm
Thread length	22 mm
Thread diameter	6 mm
Thread lead diameter	7.1 mm
Thread lead length	4.7625 mm
Tapping drill size	5

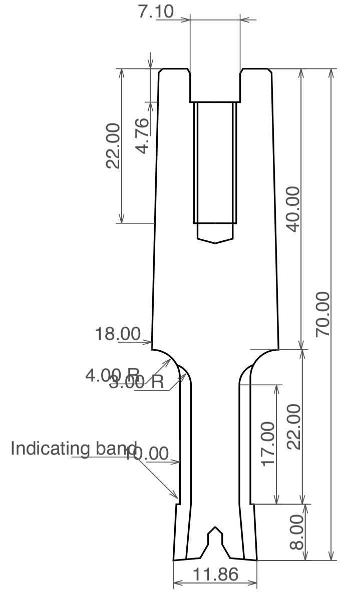
C.2 Web-based CAD/CAM system outputs for design case 3

C.2.1 Tool drawing

BASIC RACK OF CUTTER TOOTH IN CUTTING PLANE



Rack pressure angle 37.5
Chamfer pressure angle 58.1333°



TOOL DESIGN DETAILS			
PRESSURE ANGLE	37.5000	MODULE	1.7466
NUMBER OF TEETH	6	PITCH CIRCLE DIAMETER	10.4794
BASE CIRCLE DIAMETER	8.2997	CIRCULAR TOOTH TH. AT NOMINAL DIA.	2.7225
MINIMUM DEPTH	1.7167	CUTTING DEPTH	1.5500
CIRCULAR PITCH	5.4870	SIDE RELIEF ANGLE	2.3134
PROTUBERANCE	NONE	TIP RADIUS	0.1000
NOMINAL OUTSIDE DIAMETER	11.8603	SPAN OVER 2 TEETH AT NOM. DIA.	7.4511
TOP RAKE	3.0000	FRONT RAKE	5.0000
NO.2 MORSE TAPER		6mm SI THREAD	
INTERNAL PART DETAILS			
MAJOR DIAMETER MAX / MIN	17.1800 / 17.0000	MINOR DIAMETER	14.0000
MODULE	1.6000	NUMBER OF TEETH	9
CIRCULAR THICKNESS	1.6818	PRESSURE ANGLE	30.0000
T.I.F. DIAMETER	16.7000		
DESCRIPTION OF THE TOOL 1.6 MOD 30 PA 6 T DIN 5482 17 x 14		GRADE A	
DATHAN TOOL & GAUGE CO. LTD		Mean Lane, Meltham, Holmfirth, West Yorkshire, HD9 5RU, England	
DATHAN REFERENCE	931 / 113 / A / 1		
COMPONENT NO			
CUSTOMER TOOL NO SD-6			
CUSTOMER HUA-YONG MACHINE			
ROCKWELL C	66/67	MAT'L	ASP 2030
DESIGNED BY	AMA	DATE	21 AUG 2013

ALL DIMENSIONS IN MILLIMETRES

C.2.2 Design details

Design Details

All dimensions in millimetres

Inputs

Internal master gear details

Module	1.60000	Pressure angle	30.00000
No of teeth	9	Circular tooth thickness	1.68180
Major diameter max	17.18000	Minor diameter max	-
Major diameter min	17.00000	Minor diameter min	14.00000
Required T.I.F.	16.7		
Chamfer diameter max	14.60000	Chamfer angle	60.00000
Chamfer diameter min	14.55000		

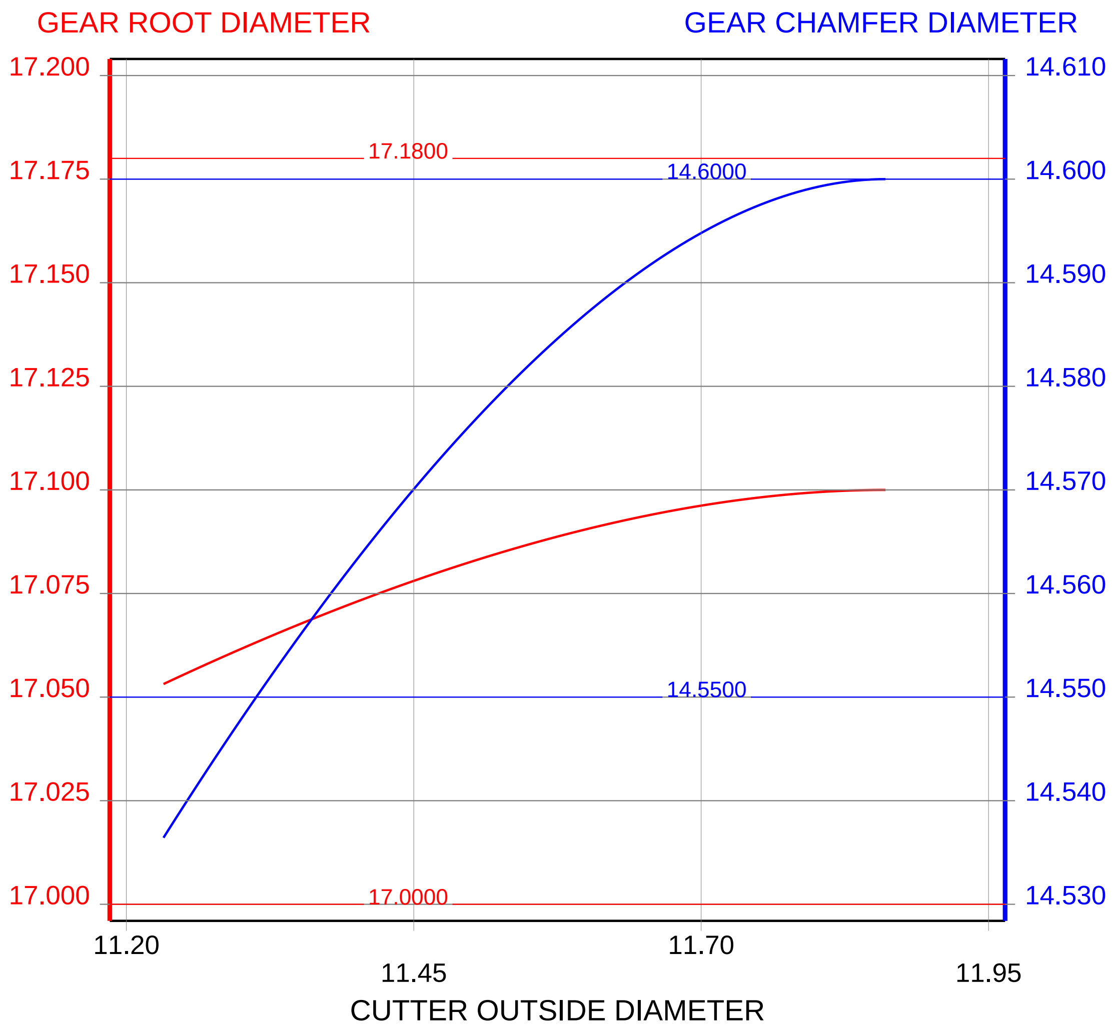
Cutter details

Module	1.74656	Design pressure angle	37.50000
No of teeth	6	Circular tooth thickness	2.72250
Front rake	5.00000	Top rake	3.00000
Oversize	0.00000	Clearance	0.25000
Life for graphs	6.00000	Height of fillet	0.10000
Rack root radius	-		
Manual root diameter	17.10000		

Results

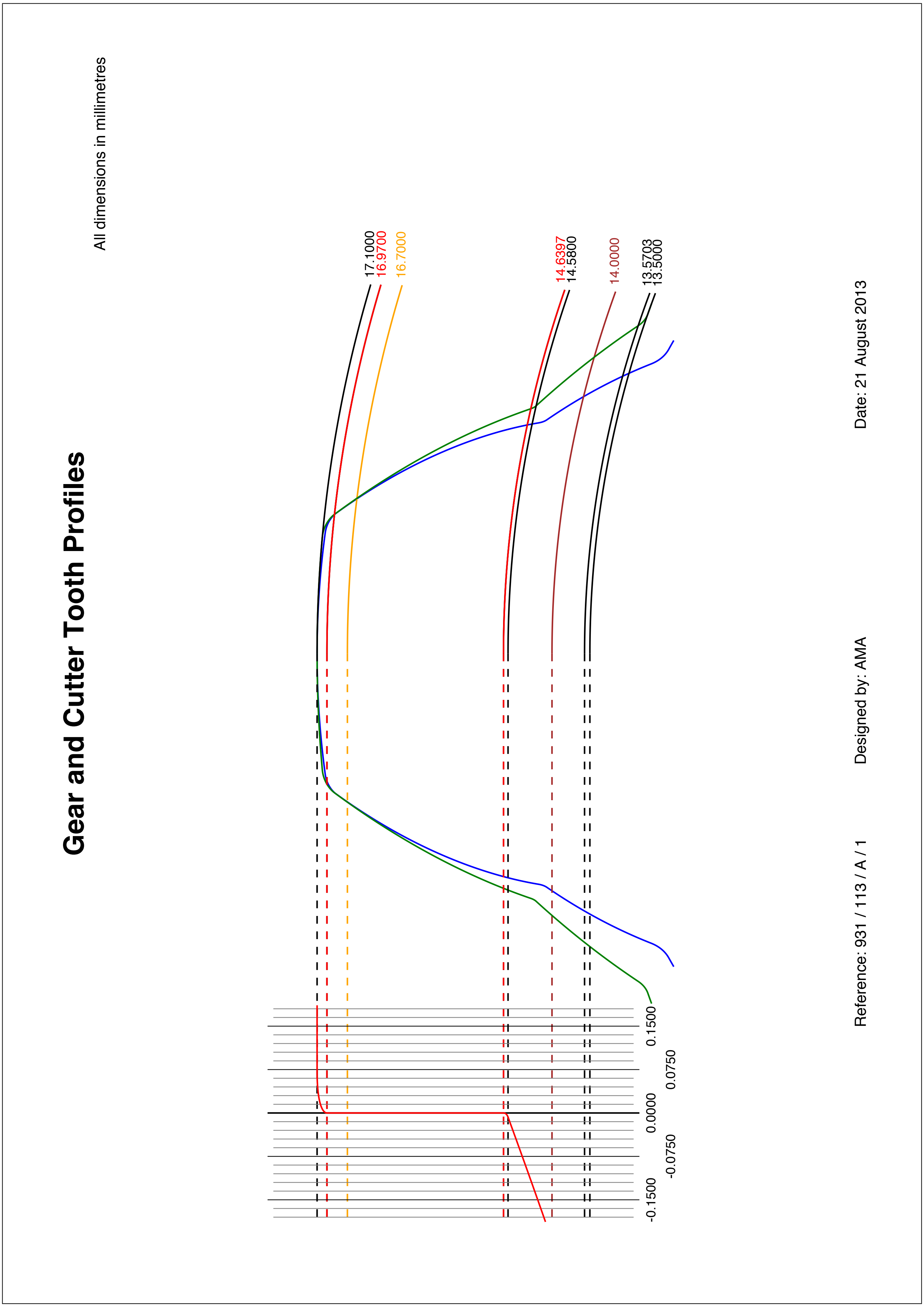
Addendum	0.69047	Dedendum	0.85953
Finished circular tooth thickness	2.72250	Pressure angle at finished OD	45.49431
Former tip thickness	1.06175	Tip thickness at finished OD	1.77009
Nominal outside diameter	11.86031	Finished outside diameter	11.86031
Former pressure angle	37.66524	Corrected pressure angle	37.62724
Correction factor	0.99405	Side relief angle	2.31343
Radius of fillet	0.25560	Height of fillet	0.10000
Max height of fillet	0.65962	Min height of fillet	0.05763
Base circle diameter	8.29966	Pitch circle diameter	10.47937

Reference: 931 / 113 / A / 1   Designed by: AMA   Date: 21 August 2013



Reference: 931 / 113 / A / 1 Designed by: AMA Date: 21 August 2013

C.2.3 Multigraph


Gear and Cutter Tooth Profiles
All dimensions in millimetres
17.1000
16.9700
16.7000
14.6397
14.5800
14.0000
13.5703
13.5000
0.1500
0.0750
0.0000
-0.0750
-0.1500
Reference: 931 / 113 / A / 1
Designed by: AMA
Date: 21 August 2013

C.2.4 Job ticket

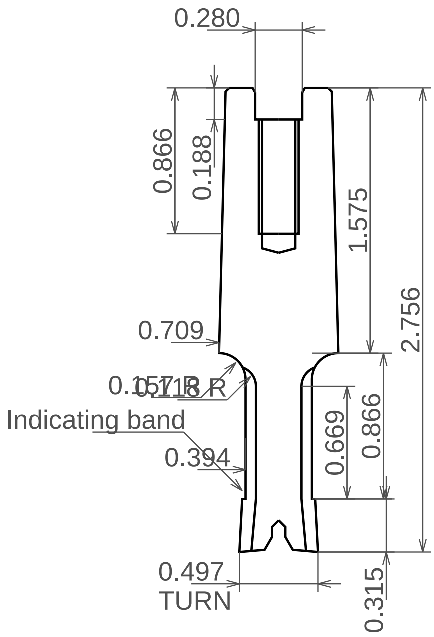
REF 931 / 113 / A **6 T**

Pitch	14.5429	PA	37.5
P.C.D.	0.4126	Min depth	0.0676
OD Theo	0.4669	OD Fin	0.4669
Theo ht Theo th	0.0341 0.1060	Theo span T spanned	0.2933 2
Fin ht Fin th	0.0341 0.1060	Fin span T spanned	0.2933 2
Front rake	5°	Top rake	3°
Cut OD			
Rotary			
Divide		Former	931/113
Fitting	No.2 Morse	Fitting flat	
Recess dia depth		Thread	6mm SI
Internal gear		Front ctr	Small
Base dia	0.3268	Circ. th.	0.1072
OD max	0.4719	OD min	0.4569
Grade	A	Tip relief	None
Tip rad	0.0039	Prot.	None
S.R.A.	2.3134	Scale	1

PONGI - HUA-YONG MACHINE

ETCH

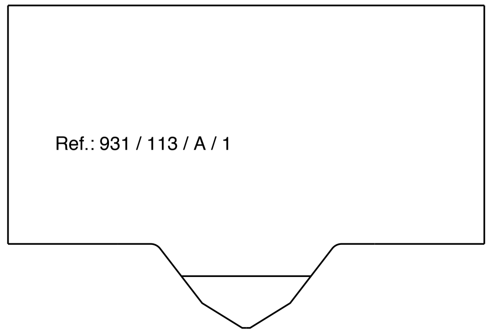
```
1.6 MOD     30 PA     6 TEETH
       DIN 5482 17 x 14
1.55 mm CD        REF 931/113
```



Date: 21 August 2013

Designed by: AMA

C.2.5 Former



HEIGHT	0.4342
THICKNESS	1.0884
ADDENDUM	0.2702
FORMER P.A.	37°39'55"
TOTAL DEPTH	0.7044
RADIUS OF FILLET	0.1006
HEIGHT OF FILLET	0.0391
FORMER SCALE	10.0
PLATE WIDTH	4.0
TOOL DIAMETER	0.1875
REFERENCE	931 / 113 / A / 1
CHAMFER P.A.	58°17'11"
HEIGHT TO CHAMFER	0.4966

C.2.6 Diaform CNC instructions

```
%
(931/113/A/1 FR=0.01006)

#5241=#5241+#7
#5247=#5247-#7

G55
T1M6
G0X-0.08344Y-0.11801C30
X-0.01300Y-0.11801
G42G1X0.00000Y-0.11801
Y-0.55000
G40X-0.01300
G0C40X-0.01300Y-0.55000
G41G1X0.00000Y-0.55000
Y-0.10801
G1X0.00000Y-0.08024
G3X-0.00173Y-0.07458R0.01011
G3X-0.00391Y-0.07226R0.00998
G1X-0.04966Y-0.03695
G1X-0.08244Y0.01611
G40X-0.09895Y0.02380
G0X-0.08344Y0.01531C30
G42G1X-0.07044Y0.01531
Y-0.01531
G40X-0.08044Y-0.01531
G0X-0.08344Y0.11801C-30
X-0.01300Y0.11801
G41G1X0.00000Y0.11801
Y0.55000
G40X-0.01300
G0C-40X-0.01300Y0.55000
G42G1X0.00000Y0.55000
Y0.10801
G1X0.00000Y0.08024
G2X-0.00173Y0.07458R0.01011
G2X-0.00391Y0.07226R0.00998
G1X-0.04966Y0.03695
G1X-0.08244Y-0.01611
G40X-0.09895Y-0.02380
G0G54
M99
%
```

Appendix D

CNC program 1 for internal master gear design

A Haas CNC program with instructions for the design with internal master gear and curved tip relief is modified using the Web-based CNC code editor and results are presented for validation.

D.1 Original CNC program 1

```
%
O504
(504/116)
(T1//.25)
(T2//.1875)
(T3//.125)
(T4//.0625)
N10T2M06
(=N190/2)
G43D2H2
M97P1000L2
M30
N1000
N20G0G20G90G54X2.5000Y-0.5000
N30G0Z-1.0000
N40S4200M3
M8
N50G1F6.00
N60G41X2.5000Y-0.5000
N70X2.5000Y0.0000
N75X0.6404Y-0.0007
N80X0.6295Y-0.0024
N85X0.6149Y-0.0062
N90X0.6019Y-0.0113
N95X0.5902Y-0.0174
N100X0.5797Y-0.0244
N105X0.5702Y-0.0321
N110X0.5591Y-0.0435
N115X0.5519Y-0.0528
N120X0.1585Y-0.6112
N125X0.1502Y-0.6230
N130X0.1434Y-0.6326
N135X0.1320Y-0.6486
```

```
N140X0.1207Y-0.6645
N145X0.1093Y-0.6805
N150X0.0978Y-0.6964
N155X0.0864Y-0.7123
N160X0.0749Y-0.7282
N165X0.0633Y-0.7441
N170X0.0518Y-0.7600
N175X-0.0518Y-0.7600
N180X-0.0633Y-0.7441
N185X-0.0749Y-0.7282
N190X-0.0864Y-0.7123
N195X-0.0978Y-0.6964
N200X-0.1093Y-0.6805
N205X-0.1207Y-0.6645
N210X-0.1320Y-0.6486
N215X-0.1434Y-0.6326
N220X-0.1502Y-0.6230
N225X-0.1585Y-0.6112
N230X-0.5519Y-0.0528
N235X-0.5591Y-0.0435
N240X-0.5702Y-0.0321
N245X-0.5797Y-0.0244
N250X-0.5902Y-0.0174
N255X-0.6019Y-0.0113
N260X-0.6149Y-0.0062
N265X-0.6295Y-0.0024
N270X-0.6404Y-0.0007
N280X-2.5000Y0.0000
N290G40X-2.5000Y-0.5000
N300G0Z0
G43T22D22H22
M99
%
```

D.2 Modified CNC program 1

```
%
O504
(504/116)
(T1//.25)
(T2//.1875)
(T3//.125)
(T4//.0625)
N10T2M06
(=N190/2)
G43D2H2
M97P1000L2
M30
N1000
N20G0G20G90G54X2.5000Y-0.5000
N30G0Z-1.0000
N40S4200M3
M8
N50G1F6.00
N60G41X2.5000Y-0.5000
N70X2.5000Y0.0000
N75X0.6404Y-0.0007
N80X0.6295Y-0.0024
N85X0.6149Y-0.0062
N90X0.6019Y-0.0113
N95X0.5902Y-0.0174
N100X0.5797Y-0.0244
N105X0.5702Y-0.0321
N110X0.5591Y-0.0435
N115X0.5519Y-0.0528
N120X0.1583Y-0.6112
N125X0.1500Y-0.6230
N130X0.1432Y-0.6326
```

```
N135X0.1318Y-0.6486
N140X0.1205Y-0.6645
N145X0.1091Y-0.6805
N150X0.0976Y-0.6964
N155X0.0862Y-0.7123
N160X0.0747Y-0.7282
N165X0.0631Y-0.7441
N170X0.0516Y-0.7600
N175X-0.0518Y-0.7600
N180X-0.0633Y-0.7441
N185X-0.0749Y-0.7282
N190X-0.0864Y-0.7123
N195X-0.0978Y-0.6964
N200X-0.1093Y-0.6805
N205X-0.1207Y-0.6645
N210X-0.1320Y-0.6486
N215X-0.1434Y-0.6326
N220X-0.1502Y-0.6230
N225X-0.1585Y-0.6112
N230X-0.5519Y-0.0528
N235X-0.5591Y-0.0435
N240X-0.5702Y-0.0321
N245X-0.5797Y-0.0244
N250X-0.5902Y-0.0174
N255X-0.6019Y-0.0113
N260X-0.6149Y-0.0062
N265X-0.6295Y-0.0024
N270X-0.6404Y-0.0007
N280X-2.5000Y0.0000
N290G40X-2.5000Y-0.5000
N300G0Z0
G43T22D22H22
M99
%
```

Appendix E

CNC program 2 for external master gear design

Haas CNC code instructions for the design with external master gear, chamfering and round fillet. The part of the tool path, that corresponds to the rounded fillet on the left side of the former profile, is corrected by +0.0004 in.

E.1 Original CNC program 2

```
%
O880
(880/97)
(T1//.25)
(T2//.1875)
(T3//.125)
(T4//.0625)
N10T1M06
G43D1H1
M97P1000L2
M30
N1000
N20G0G20G90G54X2.5000Y-0.5000
N30G0Z-1.0000
N40S3600M3
M8
N50G1F10.00
N60G41X2.5000Y-0.5000
N70X2.5000Y0.0000
N80G1X1.1324Y0.0000
N85G3X0.9233Y-0.0778R0.3198
N90G3X0.8435Y-0.1899R0.2884
N95G1X0.3084Y-1.5205
N100G1X0.0462Y-1.7533
N105G1X-0.0462Y-1.7533
N110G1X-0.3084Y-1.5205
N115G1X-0.8435Y-0.1899
N120G3X-0.9233Y-0.0778R0.2884
N125G3X-1.1324Y0.0000R0.3198
N135G1X-2.5000Y0.0000
N145G40X-2.5000Y-0.5000
N155G0Z0
G43T21D21H21
```

```
M99
%
```

E.2 Modified CNC program 2

```
%
O880
(880/97)
(T1//.25)
(T2//.1875)
(T3//.125)
(T4//.0625)
N10T1M06
G43D1H1
M97P1000L2
M30
N1000
N20G0G20G90G54X2.5000Y-0.5000
N30G0Z-1.0000
N40S3600M3
M8
N50G1F10.00
N60G41X2.5000Y-0.5000
N70X2.5000Y0.0000
N80G1X1.1324Y0.0000
N85G3X0.9237Y-0.0778R0.3198
N90G3X0.8439Y-0.1899R0.2884
N95G1X0.3084Y-1.5205
N100G1X0.0462Y-1.7533
N105G1X-0.0462Y-1.7533
N110G1X-0.3084Y-1.5205
N115G1X-0.8435Y-0.1899
N120G3X-0.9233Y-0.0778R0.2884
N125G3X-1.1324Y0.0000R0.3198
N135G1X-2.5000Y0.0000
N145G40X-2.5000Y-0.5000
N155G0Z0
G43T21D21H21
M99
%
```

Appendix F

CNC program 3 for standard cutter design

Diaform CNC program for standard design DIN 5480 (0.7). With the use of the Web-based CNC code editor the tool path coordinates, that correspond to the rounded fillet on the right side of the former profile, is corrected by -0.001 in.

F.1 Original CNC program 3

```
%
(530/117/A/1 FR=0.02357)

#5241=#5241+#7
#5247=#5247-#7

G55
T1M6
G0X-0.17730Y-0.19553C30
X-0.01300Y-0.19553
G42G1X0.00000Y-0.19553
Y-0.55000
G40X-0.01300
G0C40X-0.01300Y-0.55000
G41G1X0.00000Y-0.55000
Y-0.18553
G1X0.00000Y-0.15414
G3X-0.00495Y-0.13961R0.02380
G3X-0.01170Y-0.13368R0.02329
G1X-0.17630Y-0.03776
G40X-0.18933Y-0.04174
G0X-0.17730Y0.05676C30
G42G1X-0.16430Y0.05676
Y-0.05676
G40X-0.17430Y-0.05676
G0X-0.17730Y0.19553C-30
X-0.01300Y0.19553
G41G1X0.00000Y0.19553
Y0.55000
G40X-0.01300
G0C-40X-0.01300Y0.55000
G42G1X0.00000Y0.55000
Y0.18553
```

```
G1X0.00000Y0.15414
G2X-0.00495Y0.13961R0.02380
G2X-0.01170Y0.13368R0.02329
G1X-0.17630Y0.03776
G40X-0.18933Y0.04174
G0G54
M99
%
```

F.2 Modified CNC program 3

```
%
(530/117/A/1 FR=0.02357)

#5241=#5241+#7
#5247=#5247-#7

G55
T1M6
G0X-0.17730Y-0.19553C30
X-0.01300Y-0.19553
G42G1X0.00000Y-0.19553
Y-0.55000
G40X-0.01300
G0C40X-0.01300Y-0.55000
G41G1X0.00000Y-0.55000
Y-0.18553
G1X0.00000Y-0.15414
G3X-0.00495Y-0.13961R0.02380
G3X-0.01170Y-0.13368R0.02329
G1X-0.17630Y-0.03776
G40X-0.18933Y-0.04174
G0X-0.17730Y0.05676C30
G42G1X-0.16430Y0.05676
Y-0.05676
G40X-0.17430Y-0.05676
G0X-0.17730Y0.19553C-30
X-0.01300Y0.19553
G41G1X0.00000Y0.19553
Y0.55000
G40X-0.01300
G0C-40X-0.01300Y0.55000
G42G1X0.00000Y0.55000
Y0.18553
G1X0.00000Y0.15414
G2X-0.00495Y0.13861R0.02380
G2X-0.01170Y0.13268R0.02329
G1X-0.17630Y0.03776
G40X-0.18933Y0.04174
G0G54
M99
%
```

Appendix G

Questionnaire: the Web-based CAD/CAM system for GSCs

CAD/CAM system

In this questionnaire you will need to compare the Web-based CAD/CAM system for gear shaper cutters with the previously used desktop CAD/CAM application and provide estimates from 1 to 5 for every question, where 1 is the lowest mark and 5 is the highest mark.

1. Usability

1.1.a. Designer's productivity when using the Web–based CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

1.1.b. Designer's productivity when using the desktop CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

1.2.a. Ease of learning how to use the Web-based CAD/CAM system.
1 = very difficult; 2 = difficult; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very difficult	◯	◯	◯	◯	◯	Very easy

1.2.b. Ease of learning how to use the desktop CAD/CAM system.
1 = very difficult; 2 = difficult; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very difficult	◯	◯	◯	◯	◯	Very easy

1.3.a. Ease of use and user-friendliness of the Web–based CAD/CAM system. Consider the convenience of control elements layout, the complexity of menus, interface adjustability and personalisation features, overall behaviour of the system, how intuitive, consistent and predictable it is.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

1.3.b. Ease of use and user-friendliness of the desktop CAD/CAM system. Consider the convenience of control elements layout, the complexity of menus, interface adjustability and personalisation features, overall behaviour of the system, how intuitive, consistent and predictable it is.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

2. Interactivity

2.1.a. Level of user control over the operations executed using the Web–based CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

2.1.b. Level of user control over the operations executed using the desktop CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

2.2.a. How well the Web–based CAD/CAM system facilitates two-way communication between the user and and the software.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

2.2.b. How well the desktop CAD/CAM application facilitates two-way communication between the user and the software.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

2.3.a. Speed of processing user inputs and the response time to user requests, when using the Web–based CAD/CAM system (from the user perception point).
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excelent

2.3.b. Speed of processing user inputs and the response time to user requests, when using the desktop CAD/CAM system (from the user perception point).
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3. Reliability

3.1.a. Technical reliability (the precision and correctness) of the Web–based CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.1.b. Technical reliability (the precision and correctness) of the desktop CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.2.a. Crash-resistance of the Web-based CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.2.b. Crash-resistance of the desktop CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.3.a. Level of user confidence in the accuracy of the results, produced by the Web–based CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.3.b. Level of user confidence in the accuracy of the results, produced by the desktop CAD/CAM system.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

4. Collaboration

4.1.a. How easy it is for several people (geographically distributed) to work on the same same task using the Web-based CAD/CAM system.

1 = very difficult; 2 = difficult; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

4.1.b. How easy it is for several people (geographically distributed) to work on the same task using the desktop CAD/CAM system.

1 = very difficult; 2 = difficult; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

4.2.a. Efficiency of design activities, associated with collaboration, when using the Web-based CAD/CAM system.

1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

4.2.b. Efficiency of design activities, associated with collaboration, when using the desktop CAD/CAM system.

1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

4.3.a. Collaboration opportunities provided by the Web–based CAD/CAM system.

1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

4.3.b. Collaboration opportunities provided by the desktop CAD/CAM system.

1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

5. Scalability

5.1.a. Ease of adding new users and workplaces for the Web–based CAD/CAM system.

1 = very troublesome; 2 = troublesome; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5

	1	2	3	4	5	
Very troublesome	◯	◯	◯	◯	◯	Very easy

5.1.b. Ease of adding new users and workplaces for the desktop CAD/CAM system.
1 = very troublesome; 2 = troublesome; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very troublesome	◯	◯	◯	◯	◯	Very easy

5.2.a. Ability of the Web-based CAD/CAM system to maintain performance with the increased number of users.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

5.2.b. Ability of the desktop CAD/CAM system to maintain performance with the increased number of users.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

5.3.a. Ability of the Web-based CAD/CAM system to maintain performance, usefulness and usability regardless of expansion from concentration in a local area to a more distributed geographic pattern.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

5.3.b. Ability of the desktop CAD/CAM system to maintain performance, usefulness and usability regardless of expansion from concentration in a local area to a more distributed geographic pattern.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

6. Impact on the company

6.1. Describe the performance enhancements for the company due to the implementation of the Web-based CAD/CAM system for gear shaper cutters.

6.2. Describe the impact on company's operations from implementing the Web-based CAD/CAM system.

6.3. How is the Web-based CAD/CAM system workflow different from the previously used desktop software?

Appendix H

Questionnaire: the Web-based CNC code editor

CNC code editor

In this questionnaire you will need to compare the Web-based CNC code editor with any desktop CNC code editor that you have had experience with and provide estimates from 1 to 5 for every question, where 1 is the lowest mark and 5 is the highest mark.

1. Usability

1.1.a. User's productivity, when working with the Web-based CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

1.1.b. User's productivity, when working with the desktop CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

1.2.a. Ease of learning how to use the Web-based CNC code editor.
1 = very difficult; 2 = difficult; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

1.2.b. Ease of learning how to use the desktop CNC code editor.
1 = very difficult; 2 = difficult; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

1.3.a. Ease of use and user-friendliness of the Web-based CNC code editor. Consider the convenience of control elements layout, the complexity of menus, interface adjustability and personalisation features, overall behaviour of the system, how intuitive, consistent and predictable it is.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

1.3.b. Ease of use and user-friendliness of the desktop CNC code editor. Consider the convenience of control elements layout, the complexity of menus, interface adjustability and personalisation features, overall behaviour of the system, how intuitive, consistent and predictable it is.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

2. Interactivity

2.1.a. Level of user control over the operations executed in the Web–based CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

2.1.b. Level of user control over the operations executed in the desktop CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

2.2.a. How well the Web–based CNC code editor facilitates two-way communication between the user and the application.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

2.2.b. How well the desktop CNC code editor facilitates two-way communication between the user and the application.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

2.3.a. Speed of processing user inputs and the response time to user requests, when using the Web–based CNC code editor (from the user perception point).
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

2.3.b. Speed of processing user inputs and the response time to user requests, when using the desktop CNC code editor (from the user perception point).
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

3. Reliability

3.1.a. Technical reliability (the precision and correctness) of the Web–based CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.1.b. Technical reliability (the precision and correctness) of the desktop CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.2.a. Crash-resistance of the Web-based CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.2.b. Crash-resistance of the desktop CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.3.a. Level of user confidence in the accuracy of the results, produced by the Web-based CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

3.3.b. Level of user confidence in the accuracy of the results, produced by the desktop CNC code editor.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

4. Collaboration

4.1.a. How easy it would be for several people (geographically distributed) to work on the same task using the Web–based CNC code editor.

1 = very difficult; 2 = difficult; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

4.1.b. How easy it would be for several people (geographically distributed) to work on the same task using the desktop CNC code editor.

1 = very difficult; 2 = difficult; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very difficult	○	○	○	○	○	Very easy

4.2.a. Efficiency of CNC code editing activities, associated with collaboration, when using the Web-based CNC code editor.

1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

4.2.b. Efficiency of CNC code editing activities, associated with collaboration, when using the desktop CNC code editor.

1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

4.3.a. Collaboration opportunities provided by the Web-based CNC code editor.

1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

4.3.b. Collaboration opportunities provided by the desktop CNC code editor.

1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

5. Scalability

5.1.a. Ease of adding new users and workplaces, when using the Web–based CNC code editor.

1 = very troublesome; 2 = troublesome; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5

Very troublesome ◯ ◯ ◯ ◯ ◯ Very easy

5.1.b. Ease of adding new users and workplaces, when using the desktop CNC code editor.
1 = very troublesome; 2 = troublesome; 3 = moderate; 4 = easy; 5 = very easy.

	1	2	3	4	5	
Very troublesome	◯	◯	◯	◯	◯	Very easy

5.2.a. Ability of the Web-based CNC code editor to maintain performance with the increased number of users.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

5.2.b. Ability of the desktop CNC code editor to maintain performance with the increased number of users.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

5.3.a. Ability of the Web-based CNC code editor to maintain performance, usefulness and usability regardless of expansion from concentration in a local area to a more distributed geographic pattern.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

5.3.b. Ability of the desktop CNC code editor to maintain performance, usefulness and usability regardless of expansion from concentration in a local area to a more distributed geographic pattern.
1 = poor; 2 = fair; 3 = good; 4 = very good; 5 = excellent.

	1	2	3	4	5	
Poor	◯	◯	◯	◯	◯	Excellent

6. Impact on the company

6.1. Describe the performance enhancements for the company due to the implementation of the Web-based CNC code editor for gear shaper cutters

6.2. Describe the impact on company's operations from implementing the Web-based CNC code editor.

6.2. How is the Web-based CNC code editor workflow different from the previously used desktop CNC code editing software?