International Journal of Science and Engineering Applications
Volume 3 Issue 6, 2014, ISSN-2319-7560 (Online)

A Methodology for Developing Web-based CAD/CAM systems

Anna Malahova
Manufacturing and Materials Department
Cranfield University
Bedfordshire, United Kingdom

Ashutosh Tiwari
Manufacturing and Materials Department
Cranfield University
Bedfordshire, United Kingdom

Abstract: 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.

This research establishes a methodology for creating Web-based CAD/CAM software systems, which will help develop complex Web-based CAD/CAM systems to industrial quality standards in a time and cost effective manner. The crucial parts of the methodology are a novel project development model facilitating architecture optimisation early in the project and a novel approach for planning based on time reserve management and task prioritisation.

Keywords: methodology; Web-based; CAD/CAM; software model; development; reserve-based planning

1. INTRODUCTION

Today, the Web is used as a platform supporting many business areas and can provide significant advantages for CAD/CAM software development too. Web-based approach allows reducing operating costs and improving agility, responsiveness and overall competitiveness of companies. At the same time, the task of Web-based CAD/CAM software development has a number of associated challenges, such as Internet connectivity and security, performance optimisation, collaboration and interactivity requirements [1-5]. Therefore, to exploit state-of-the-art Web technologies in the area of CAD/CAM efficiently a systematic approach is required.

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 [6]. The development methodology and the design of Web-based CAD/CAM applications are usually based on trial and error [3, 7].

Introducing a specialised methodology for Web-based CAD/CAM development can 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.

A methodology for creating Web-based CAD/CAM software can be explored and formalised by identifying features specific to Web-based CAD/CAM software development and by comparing those against known software development methods and positive software engineering experiences [8].

Web-based CAD/CAM software combines the ability to work in the World Wide Web (WWW) inherent to Web applications, application of accumulated knowledge for solving engineering and production problems characteristic to CAD/CAM systems, as well as possibilities for analysis and knowledge capture in a particular field associated with scientific CAD/CAE and Computational Science and Engineering (CSE) applications. Development approach is different for each of these application types.

2. RATIONALE AND PREREQUISITES FOR SPECIALISED METHODOLOGY

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.

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 tends to cause difficulties in introduction of new features and technologies, usually requiring significant changes in software architecture [9]. Scope, understanding and volatility problems are relevant for Web-based CAD/CAM software requirements elicitation [10].

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 [11]. 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 [12], but it brings in a risk that learning can overweight the actual development and 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. 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 [13, 14].

2.1 Features specific to Web-based CAD/CAM software

Every software development model can be successfully applied to a limited scope of projects, obeying certain conditions [15, 16]. Approaching Web-based CAD/CAM system development using one of existing software methodologies is troublesome due to the following combination of characteristics specific to Web-based CAD/CAM software:

1. Requirements containing specific industry-related and often science-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;

2.2 Web-based CAD/CAM software development challenges

The features specific to Web-based CAD/CAM software subsequently lead to a set of challenges associated with development of 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.

2.3 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, 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.
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.

3. METHODOLOGY DESCRIPTION

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 1. Key principles for the new methodology are derived from the features specific to Web-based CAD/CAM software considering the positive software engineering experiences [8]. 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.

For addressing Web-based CAD/CAM system development challenges summarised in Section 2.2 the following key principles should be kept in mind during the development:

1. User involvement is vital for accurate and timely feedback on the project development.
2. The development process is iterative and incremental and focused on frequent delivery based on task prioritising.
3. 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.
4. Big and complex tasks should be broken down to smaller parts that are easier to deal with.
5. Exploratory requirement identification should be applied to eliminate ambiguity and deal with requirement uncertainty.
6. Throwaway prototyping, design patterns and refactoring are used to ensure the quality of the software design.

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

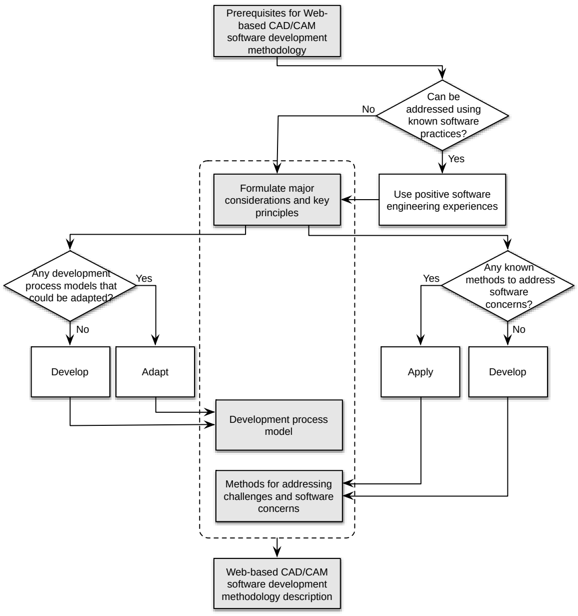

Figure 1. Steps and decisions used to construct a methodology for Web-based CAD/CAM development

4. DEVELOPMENT PROCESS

The proposed software life-cycle builds on features of other widely used models and extends them to address the methodology prerequisites formulated in Section 2.3.

The methodology consists of the following sequential phases:

1. 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.
2. 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.
3. 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.

The project development process model employed by the methodology is shown in Figure 2. 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.

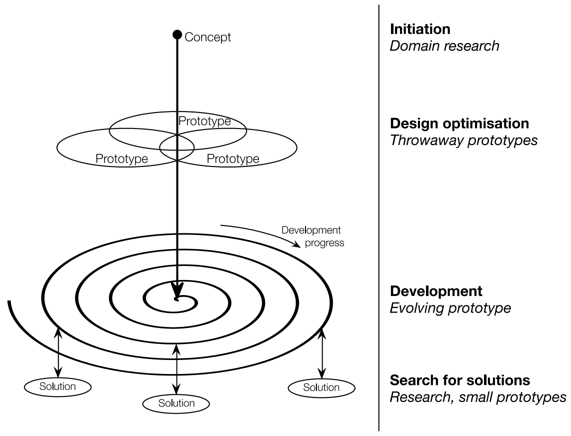

Figure 2. Software development process model

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.

Improved quality of initial software design allows eliminating the necessity of making critical design changes during the incremental development. 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: 1) Refining requirements; 2) Extending design documentation; 3) Assessing risks; 4) Prioritise tasks; 5) Short-term planning and adjusting time and effort estimation; 6) Refactoring; 7) Implementation; 8) Testing and fixing discovered errors; 9) Integration with the production system; 10) Obtaining feedback.

In addition the project can require research activities or exploratory development 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.

4.1 Planning

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

1. 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) [17, 18] is limited to the completeness and correctness of identified critical chains.
2. 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.
3. 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 time 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 supposed 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 3 shows task composition for each iteration depending on the intensity of using reserved effort.

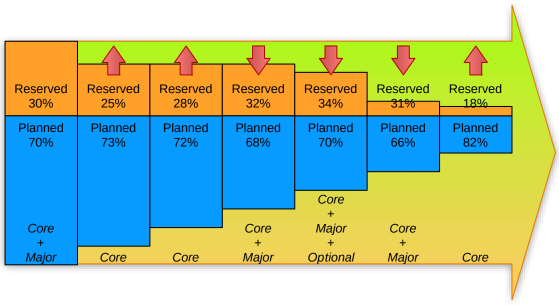

Figure 3. Planning task composition for each iteration depending on the intensity of using reserved effort

Normally the priority of project activities would be the following: 1. Reduce uncertainty; 2. Implement complex tasks; 3. Deliver business value; 4. Refine and improve the code.

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 can altered for different project phases, as shown in Table 1.

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.

Table 1. 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

Tasks from one, two or all three given categories can be scheduled for implementation within single project iteration. The composition of tasks for 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 4.

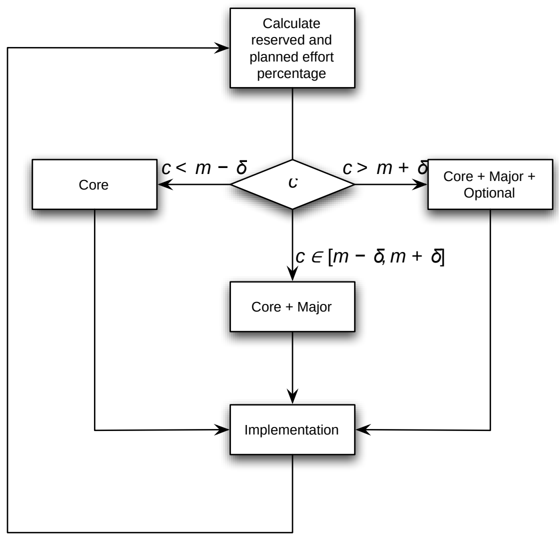

Figure 4. A block scheme showing the process of decision making about the task composition for each iteration. *c* - current reserve; *m* - reserve threshold; δ - lag size.

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 3. with red arrows.

5. APPLICATION AND VALIDATION

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 sponsoring company and incorporate the development of a Web-based CAD/CAM system for involute spur gear shaper cutters (Figure 5) and a Web-based CNC code editor for online modification of the profile for manufacturing gear shaper cutters (Figure 6). 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.

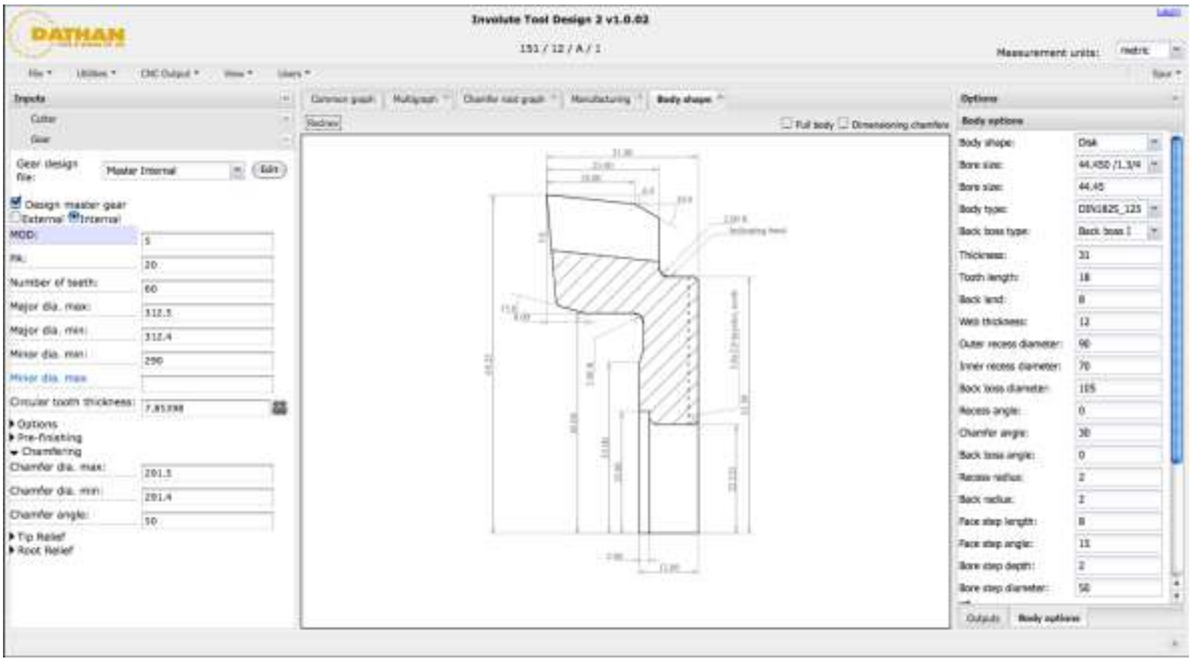

Figure 5. Web-based CAD/CAM system for gear shaper cutters

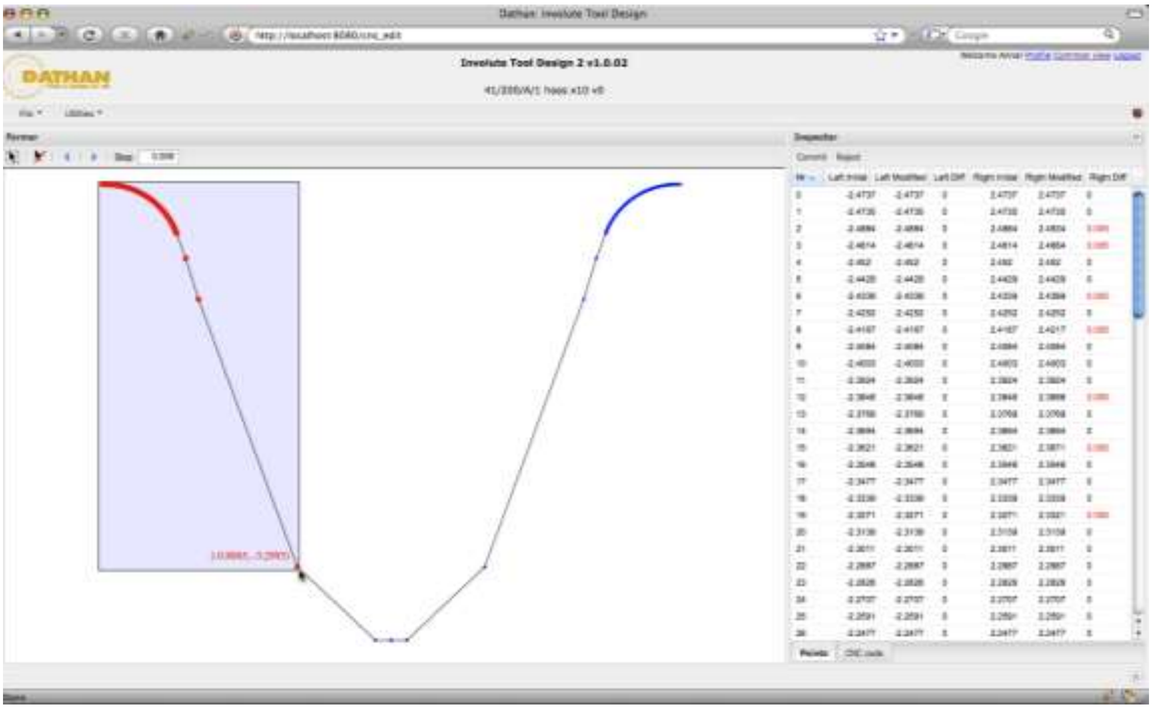

Figure 6. Web-based CNC code editor for online profile modification

The development of case studies using the established methodology resulted in on-time delivery of two industrial Web-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. Feedback from industry representatives indicates that usability, interactivity and reliability estimates for Web-based CAD/CAM system and

CNC editor are equally good or slightly better than those for desktop software used by the company previously, and unlike the desktop software the developed Web-based CAD/CAM systems received high collaboration and scalability estimates (Figures 7–8).

The development of case studies 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 case studies.

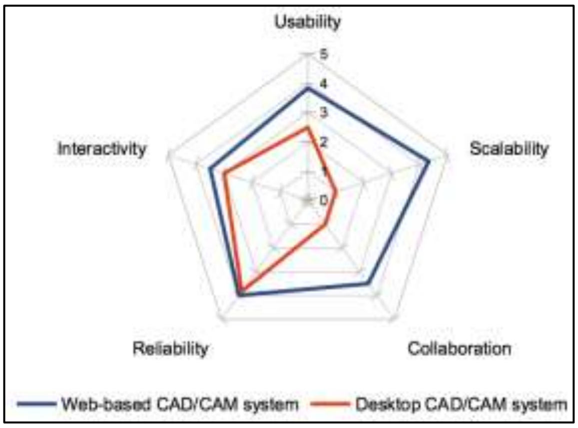

Figure 7. Evaluation of the Web-based CAD/CAM system for gear shaper compared to similar desktop software based on feedback from industry representatives

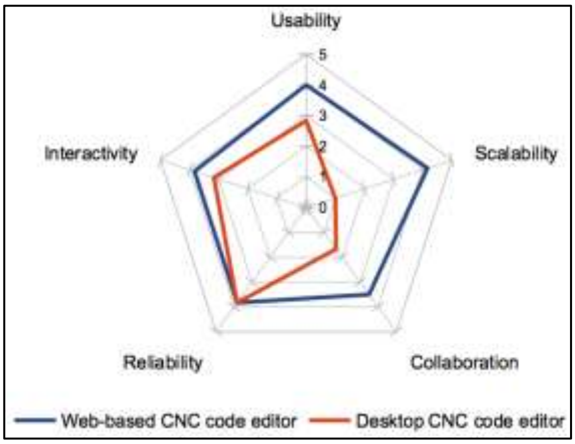

Figure 8. Evaluation of the Web-based CNC code editor compared to similar desktop software based on feedback from industry representatives

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. 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 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, which is 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.

6. APPLICABILITY AND PROJECTED BENEFITS

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.

Applying the proposed methodology for Web-based CAD/CAM system development can 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.

7. CONCLUSIONS

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 practices 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.

The research mentions characteristics specific to Web-based CAD/CAM software development that make the application of the existing methodologies troublesome. These characteristics include ambiguous and changing requirements, which usually contain specific industry related and often scientifically-intensive information, multi-disciplinary teams, stretched over time development and continuous evolution of the application, changing technologies and high level of uncertainty.

The methodology addresses challenges governed by the features specific to industrial Web-based CAD/CAM applications and provides techniques that enable effective and quality software development.

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 time reserved for overcoming uncertainty. The reserve is supposed 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 methodology description assists a developer in identifying if the established methodology could be applied in a particular project, what benefits it can bring and what steps to undertake to apply the methodology.

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.

The proposed methodology was applied on real software development case studies incorporating a Web-based CAD/CAM system for gear shaper cutters 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 used in production. Finally, the case studies demonstrate advantages of created Web-based CAD/CAM applications compared to similar desktop software.

8. REFERENCES

[1] 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.

[2] 2012 trends report: Application security risks, Tech. rep., Cenzic, Inc. (March 11 2012).

[3] 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.

[4] T. X. L. N. X. B. Cai, H.a, Framework of geometry modeling integrated platform based on services for large-scale information visualization, Journal of Multimedia 5 (4) (2010) 361–368.

[5] 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.

[6] [89] H. Yang, D. Xue, Recent research on developing web-based manufacturing systems: A review, International Journal of Production Research 41 (15) (2003) 3601–3629.

[7] [90] 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.

[8] 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.

[9] 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.

[10] M. G. Christel, K. C. Kan, Issues in requirements elicitation, Tech. rep., Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Pennsylvania 15213 (September 1992).

[11] K. W. Wood, W.A., Exploring XP for scientific research, IEEE Software 20 (3) (2003) 30–36+4.

[12] D. b. Sotirovski, Heuristics for iterative software development, IEEE Software 18 (3) (2001) 66–73.

[13] J. Highsmith, Extreme programming, Tech. rep., Agile Project Management Advisory Service (2000).

[14] B. K. Jayaswal, P. C. Patton, Design for Trustworthy Software: Tools, Techniques, and Methodology of Developing Robust Software, 1st Edition, Prentice Hall, 2006.

[15] D. Green, A. DiCaterino, A survey of system development process models, Tech. rep., Center for Technology in Government, University at Albany / SUNY (February 1998).

[16] Selecting a development approach (2005), https://www.cms.gov/systemlifecycleframework/downloads/selectingdevelopmentapproach.pdf, (accessed: 15 September 2011).

[17] E. M. Goldratt, Critical Chain. A Business Novel, North River Press, 1997.

[18] L. P. Leach, Critical chain project management, Artech House technology management and professional development library, Artech House, 2000.