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.

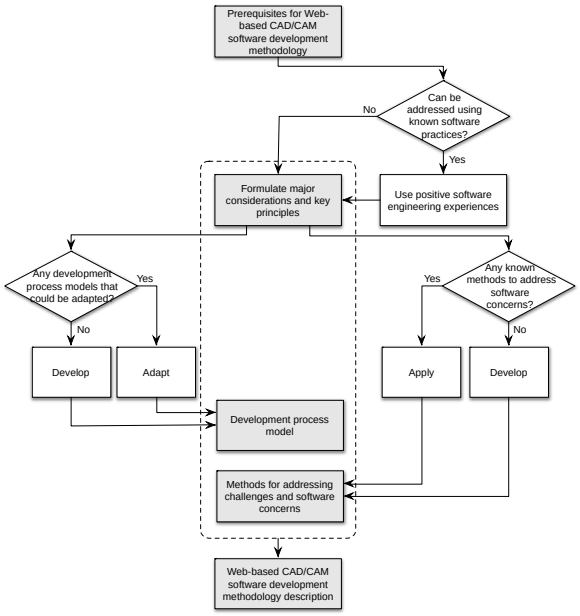

Figure 1. Steps and decisions used to construct a methodology for Web-based CAD/CAM development

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.

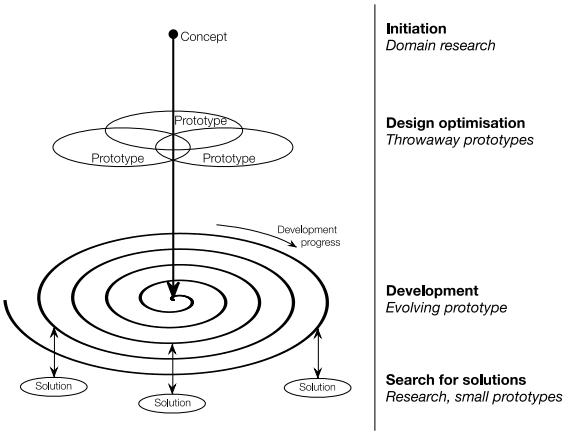

Figure 2. Software development process model

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.

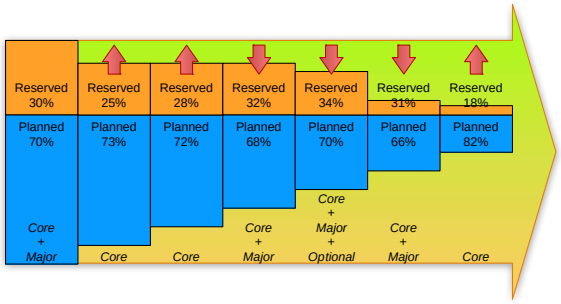

Figure 3. Planning task composition for each iteration depending on the intensity of using reserved effort

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.

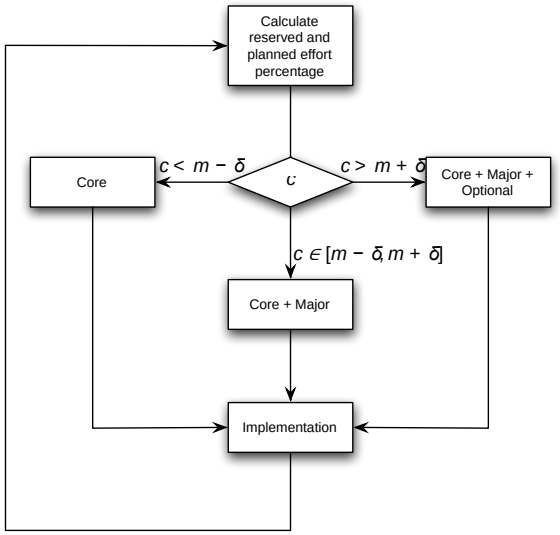

Figure 4. A block scheme showing the process of decision making about the task composition for each iteration. *c* - current reserve; *m* - reserve threshold; *δ* - lag size.

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.

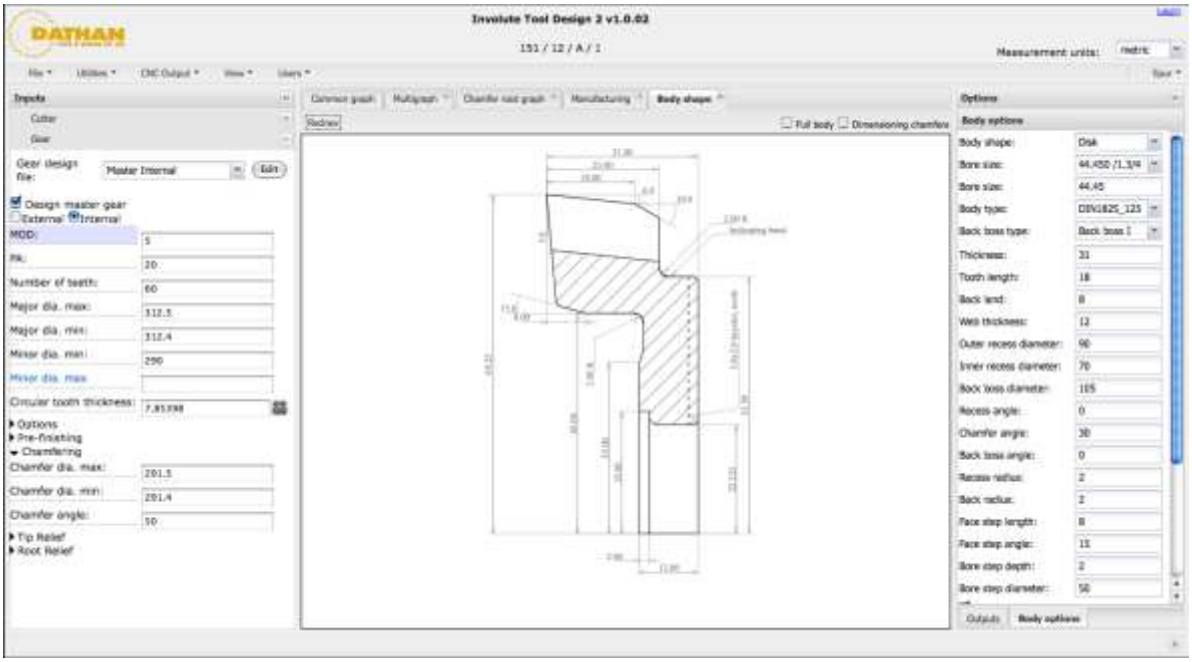

Figure 5. Web-based CAD/CAM system for gear shaper cutters

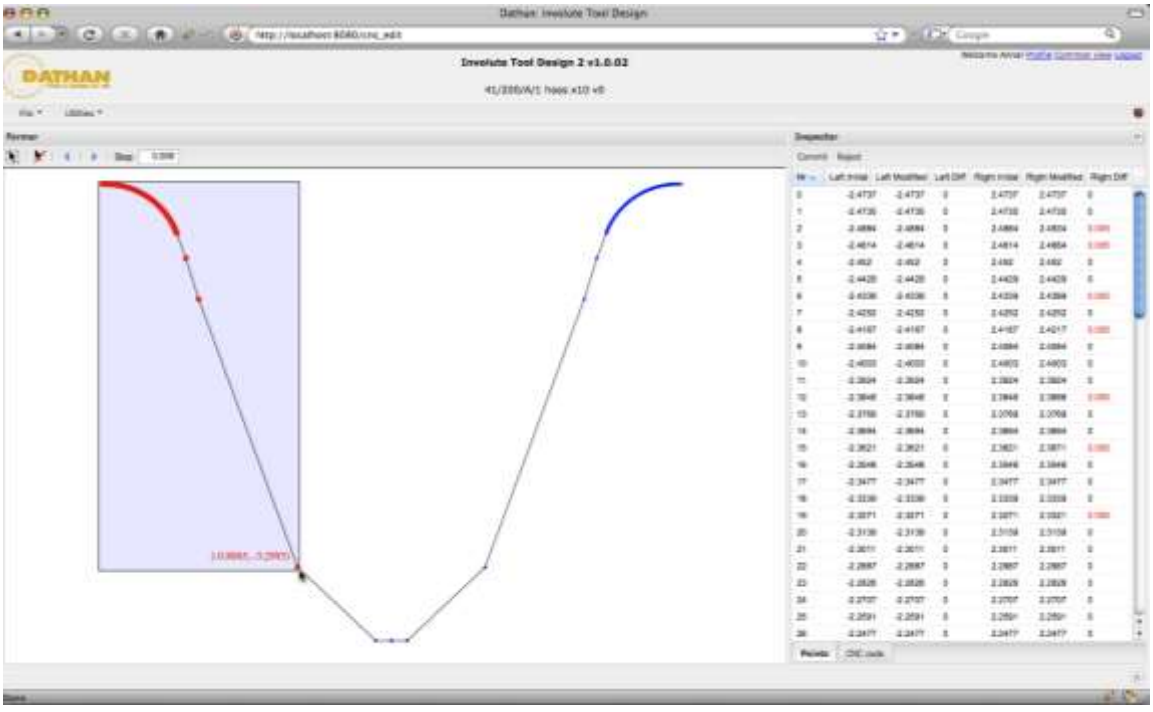

Figure 6. Web-based CNC code editor for online profile modification

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.

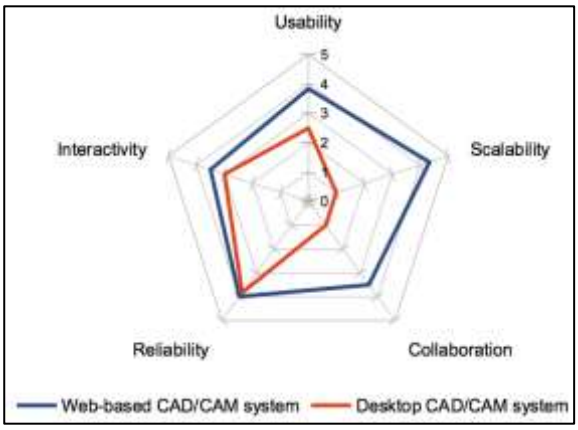

Figure 7. Evaluation of the Web-based CAD/CAM system for gear shaper compared to similar desktop software based on feedback from industry representatives

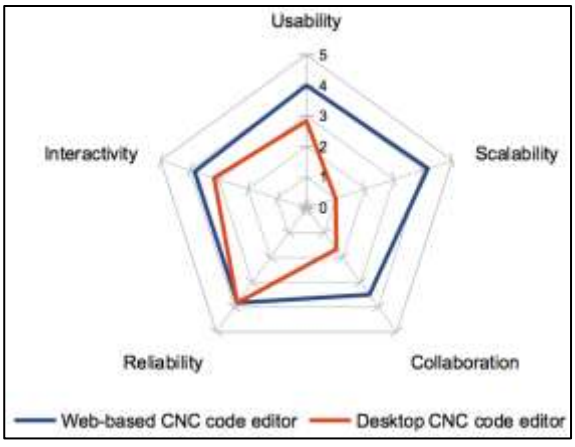

Figure 8. Evaluation of the Web-based CNC code editor compared to similar desktop software based on feedback from industry representatives

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.

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

A New Group Signature Scheme with Efficient Membership Revocation

Thu Thu Mon Oo
University of Technology
(Yatanarpon Cyber City)
Pyin Oo Lwin, Myanmar

Abstract: In group signature schemes, the members of the group are allowed to sign messages anonymously on the behalf of the group. In this case, other group members and the outsiders from the group cannot see which member signed the messages. The organizational structure which should support the safety of privacy may need to provide a degree of anonymity to the individuals conducting the transactions. Moreover, the current methods of revocation property of the group signature scheme do not revoke to allow valid signature under an old secret key of the group manager. And it is remaining as a challenge to be independent on the size of the group public key when the group size is increasing. For this above facts, this paper will be proposed to achieve anonymous revocation based on the concept of group signature more effectively.

Keywords: Cryptograph, Digital Signature, Group Digital Signature, Anonymity, Revocation

1. INTRODUCTION

With the rapid improvement of technology, digital signatures have become as an important role in the aspects of information security, identity authentication, data integrity, anonymity, message authentication, and so on.

Group signatures are one kind of digital signatures with special efforts. This signature can allow any group members to sign message on behalf of the group while remaining anonymity. When the case of disputes will be occurred, the group manager opens the signatures to revoke anonymity of group members. Group signatures can be applied with most of the activities of electronic polities and electronic commerce such as e-voting, e-bidding, e-cash, e-banking and so forth. It is more suitable for some application in which desirable to hide organizational structure.

Most group signatures can usually be used to conceal the internal structures of the group. The security and efficiency of most group signature schemes proposed previously, are not very ideal. In some previous group signature schemes, they cannot protect adversaries from colluding attacking or universally forging. As a result, they cannot serve the whole advantage of groups. And then most of group signature schemes cannot delete group members effectively so that they cannot meet the needs of dynamic groups in reality.

The group signatures are a "generalization" of the credential/ membership authentication schemes, in which one person proves that he belongs to a certain group. It has the following properties:

- only members of the group can sign messages
- the receiver can verify that it is a valid group signature, but cannot discover which group member made
- if necessary, the signature can be opened, so that the person who signed the message is revealed

Some schemes were described to split the functions of the group manager as two manages. The authorities and tasks of the group manager are divided into two parts in this proposed scheme. This desirable fact allows the distribution of the trust. In the proposed signature scheme, the membership certificate is in fact the zero-knowledge for others except one of the group managers who can add new members. To design a group signature scheme profitably, it is still an open problem to be secure and efficient.

In this proposed scheme, the group digital signature with distributed authorities can be used to decrease the managing workloads of the managers in the group. The members can be added or removed from the group as a result of join and revoke algorithms. The proposed scheme may be supported the properties of anonymity and revocation particularly. While generating the group signature, the requested message from one of group members can only be signed with the private key of issuing manager. Furthermore, no knowledge of the information about the members in this signature can be given. In the case of cheating, this member who cheated must be revoked by the opening manager anonymously, and was made to be unable to sign in the future. With improving awareness about security of the group, how to protect the source of the signature and confirm the integrity of the transmitted message from the view of Outsider is an important issue.

The rest of this paper is organized as follows. In Section 2, the related work of group signature schemes will be described. Section 3 describes the proposed group signature scheme. Section 4 introduces the procedures of the proposed signature scheme. And preliminaries for analysis of the proposed scheme will be expressed in Section 5. In Section 6, the security analysis of the proposed scheme will be expressed. Finally, the paper concludes in Section 7.

2. RELATED WORK

To conceal organizational structures group signature can be used. For example, an employee of a company can use group signature to sign document on behalf of the company. In this situation, it is sufficient for a verifier to know that some behalf of the company has signed. Verifier does not need to check whether the employee is allowed to sign document on behalf of the company.

Killan and Petrank [3] also indicate the concept of separability. That is, if the group manager is split into a membership manager and a revocation manager, the revocation manager and the membership manager work in

concert to open the identity of the signer. But they did not propose any group signature scheme to achieve this function.

Bellare, Shi and Zhang [4] strengthened the security model to include dynamic enrollment of members. This security model also separated the group manager's role into two parts: issuer and opener. The issuer is responsible for enrolling members, but cannot trace who has signed a group signature. The opener on the other hand cannot enroll members, but can open a group signature to see who signed it. More-over, it was required that this opener should be able to prove that said member made the group signature to avoid false accusations of members. [5] demonstrated that trapdoor permutations suffice also for constructing group signatures in this model. Both of these schemes use general and complicated primitives and are very inefficient.

Jan Camenisch presented an efficient group signature scheme in [6]. Providing computational anonymity, ability to add (or remove) group members after the initial setup, and the possibility of being generalized by allowing authorized set of group members to sign collectively on behalf of the group. This scheme can be extended to allow the functionality of the group manager to be shared among several entities. The drawbacks include the size of the public key and the signature size both which are proportional to group size.

In [7], most previous results are shown the following disadvantages: the size of the group public and the length of the group signature depend on the size of the group. And then new member addition requires restarting the entire system or involves re-issuing all members' keys and changing the group public key. Furthermore, revocation of group members also requires that.

3. PROPOSED GROUP SIGNATURE SCHEME

The entities of the group involved in this proposed scheme are:

- The central manager (CM) will establish the group with the whole authority and is served as the trusted party. It can be performed as the owner of the group. CM controls the group responsibilities that are divided into two authorities: the issuing manager (IM) and the opening manager (OM), and then CM creates all of the group keys.
- The issuing manager (IM) will produce the signature on the behalf of the group. IM has read –only access to the group storage for the authentication of the member to sign the message. Although IM can only be signed the message to create group signature with his private key as the group private key, IM cannot know the content of the message that is to send to outsider.
- The opening manager (OM) can add new members of the group, and has both read and write accesses to the group storage. OM can also generate the membership certificates to group members for joining and revoking processes.
- Members of the group, who have already joined the group and accepted the membership certificates, will send the message for producing the signature to IM. The group members create the hash value of the original message that sends to the outsider for the authentication of the message.
- The outsiders do not belong to the group but have the access to the public key of group to verify the group signature.

In this proposed scheme, the results that can be obtained are the efficiency of revoking the members and signing the group signature for some special tasks of anonymity and revocation. Because the group members cannot sign themselves on the behalf of the group, this proposed scheme will be given the advantage that the cheated member cannot use the abilities of signing the message in this group. The collaboration of the group member and OM can generate the valid group signature in the proposed scheme. As a result, it can be implemented with better improvements in the signing procedure.

In the event of dispute, OM extracts the identity of the member from the membership certificate in the signature, and revokes this member from the group. Although OM do not reveal the identity of this member to the outsider, he sends the notified message that is not a valid signature at the current time and suggest to discard the corresponding message of this signature. But this notified message has no knowledge of the information about the originator of signature for protecting the privacy of the group member. In the case of cheating, this member who cheated must be revoked by the opening manager anonymously, and was made to be unable to sign in the future. With improving awareness about security of the group, how to protect the source of the signature and confirm the integrity of the transmitted message from the view of Outsider is an important issue.

4. PROCEDURES OF PROPOSED SCHEME

In the proposed group signature scheme, it will be provided the secrecy of the group identities of new members effectively by encrypting their own NRC numbers that can be known themselves only. The procedures of the proposed group signature scheme are:

1. Setup
2. Join
3. Sign
4. Verify
5. Open
6. Revoke

The above six steps of the proposed scheme are described briefly as follow:

In setup algorithm, CM runs to generate the key pair of the group, and the secret key for membership certificates. The private key of the group is assigned as the secret key of IM and the remaining one from the key pair is also assigned as the group public key. The secret key to produce the membership certificate is used as the own key of OM.

Join algorithm is required to check the existence of the member whether already joined the group or not. If the new one is not a member, OM will add all of the data of this one to the group storage. Then OM sends the membership certificate and member identity (M_ID) to this new member, and updates the group storage. OM encrypts the required data of the member with his private key as the group certificates, and like this way, OM encrypts the group ID with the NRC number of new member. And the members can obtain the group ID by decrypting with their corresponding NRC number. Members do not know anything about their certificates.

The member computes the member identity with his NRC number to get his group ID. Before the member does not accept the group signature, the member who wants to sign the message produces the hash value of the message to send to the outsiders. After hashing the message, the member sends member ID, hash value of message and his membership certificate to IM. When IM receives the request to sign, IM checks member ID that is compared as the same one in the group storage. If it is already existed, IM will produce the signature with his secret key by encrypting both the membership certificate and hash value, and send this signature to the group member who sent the request to produce the signature. The member sends the original message, the hash value of this message and the signature by concatenating to the outsider.

According to the verify algorithm, the verifier or outsider can verify the signature with the group public key. After verifying the signature, he extracts the hash value of the original message from the signature, and compares both of the hash value what he extracts. If they are all the same, the outsider will accept as the valid signature and the authentication of the message. Unfortunately if they cannot verify that, they will send the signature to OM to verify.

In the case of cheats or disputes, the open algorithm will be ran by OM and returned the identity of the member who signs the corresponding message. In this event, OM will reply to the outsider what the message is the valid signature or not.

After OM got the member ID from the membership certificate of this member, OM deletes the record of the identity of member from member list and adds this record of deleted member to Blacklist of member.

5. PRELIMINARIES

Some cryptographic assumptions are reviewed that are intended to satisfy some security properties of proposed group signature scheme.

5.1 Discrete Logarithm Problem

Discrete logarithms are logarithms defined with regard to multiplicative cyclic groups. If G is a multiplicative cyclic group and g is a generator of G, then from the definition of cyclic groups, it is known that every element h in G can be written as g^x for some x. The discrete logarithm to the base g of h in the group G is defined to be x. For example, if the group is Z_5^*, and the generator is 2, then the discrete logarithm of 1 is 4 because $2^4 \equiv 1 \bmod 5$.

The discrete logarithm problem is defined as: given a group G, a generator g of the group and an element h of G, to find the discrete logarithm to the base g of h in the group G. Discrete logarithm problem is not always hard. The hardness of finding discrete logarithms depends on the groups. For example, a popular choice of groups for discrete logarithm based crypto-systems is Z_p^* where p is a prime number.

5.2 Computational Diffie-Hellman Assumption

The computational Diffie–Hellman (CDH assumption) is the assumption that a certain computational problem within a cyclic group is hard.

Consider a cyclic group G of order q. The CDH assumption states that, given (g, g^a, g^b). For a randomly chosen generator g and random $a,b \in \{0,\dots, q-1\}$, it is computationally intractable to compute the value g^{ab}.

5.3 Decisional Diffie-Hellman Assumption

The decisional Diffie–Hellman (DDH) assumption is a computational hardness assumption about a certain problem involving discrete logarithms in cyclic groups. It is used as the basis to prove the security of many cryptographic protocols.

Consider a (multiplicative) cyclic group G of order q, and with generator g. The DDH assumption states that, given g^a and g^b for uniformly and independently chosen $a,b \in Z_q$, the value g^{ab} "looks like" a random element in G.

This intuitive notion is formally stated by saying that the following two probability distributions are computationally indistinguishable:

- (g^a, g^b, g^{ab}), where a and b are randomly and independently chosen from Z_q.
- (g^a, g^b, g^c), where a, b and c are randomly and independently chosen from Z_q.

6. SECURITY ANALYSIS OF PROPOSED SCHEME

An overview of the proposed group signature scheme will be defined as follow. This proposed scheme is based on the above mentioned assumptions. The symbol || denotes the concatenation of two binary strings.

6.1 Setup

The setup procedure is as follow. The Central Manager of the group must perform the following steps:

1. Chooses random primes p, q and then the central manager computes an RSA modulus $n=pq$.
2. Chooses a public exponent e randomly such that e is relatively prime, and compute $d=e^{-1} \bmod (p-1)(q-1)$.
3. Selects an element g of Z_n^* of order n. Let G be a cyclic subgroup of Z_n^*.
4. Chooses a secret key $X \in G$ randomly.
5. Finally, a collision-resistant hash function H: $\{0,1\}^* \rightarrow \{0,1\}^k$.

The public key of the group is $P= (n, e, g)$ and the secret key of the Issuing Manager is $S = (n, d)$.

6.2 Join

In the proposed scheme, if a user wants to join the group, the scheme is assumed that the communication between the group member and the group manager as the Opening Manager is secure. To obtain the membership certificate of the group, each user must perform the following steps with the Opening Manager.

1. The user U_i selects an element x_i as his own identity (ID), and sends x_i and his data D_i.
2. The Opening Manager checks x_i from the group storage. If it is not in the group database, he selects random

number (id_i) for group ID, and computes the membership ID (ID_i) and the group certificate C_i.

- $ID_i = x_i.id_i$
- $C_i = X.(D_i \,||\, id_i)$.

3. The Opening Manager sends ID_i and C_i to user U_i. After the user U_i gets the pair (ID_i, C_i), he can be as a member of the group.
4. The user U_i computes $id_i' = x_i^{-1}.\ ID_i = x_i^{-1}.\ x_i.id_i$. The pair ($id_i$, C_i) is the membership certificate of the new member.

Consequently, at the end of the steps, the Opening Manager creates a new entry in the group database and stores id_i and D_i in the new entry. And group certificates cannot be identified by anyone except the Opening Manager because this group certificates can be issued with the secret key that is known only by the Opening Manager. As a result, the proposed group signature scheme can be satisfied with anonymity property.

6.3 Sign

A group member U_i with a membership certificate (id_i, C_i), can generate group signatures with the Issuing Manager on a message m as follows:

1. The member U_i computes $H(m)$, the hashed value of the message, and he sends $(id_i \,||\, C_i \,||\, H(m))$ to the Issuing Manager to issue the group signature.
2. The Issuing Manager gets id_i and compares whether this member is valid or not from the group database.
3. If this member is valid, the Issuing Manager computes the signature σ and sends the signature to the member.

- $\sigma = (C_i \,||\, H(m))^d \bmod n$.

4. The member U_i sends the group signature with the message $(\sigma \,||\, m \,||\, H'(m))$ to Outsider.

The Issuing Manager can issue the signature σ if the member who has the validity of the existence in the group is true by checking his group ID. Therefore the proposed signature can be identified that has the property of unforgeability.

6.4 Verify

The resulting signature $(\sigma \,||\, m \,||\, H'(m))$ of a message m can be verified by the Outsider as follows:

1. Computes $\sigma^e \bmod n = (((C_i \,||\, H(m))^d)^e \bmod n = C_i \,||\, H(m)$.
2. Accept the group signature $(\sigma \,||\, m \,||\, H'(m))$ if and only if $H'(m) = H(m)$.

As a result, if the Outsider accepts what the group signature is valid after Verify algorithm, the proposed group signature is satisfied with the property of correctness.

6.5 Open

Unless the Outsider can verify the group signature, he resends the signature σ to the Opening Manager to check whether the originator of the signature is valid or not. In the case of dispute, the Opening Manager can find out which one of the group members issued this signature and perform the following steps:

1. Computes $\sigma^e \bmod n = (((C_i \,||\, H(m))^d)^e \bmod n = C_i \,||\, H(m)$.
2. Computes $X^{-1}.C_i = X^{-1}.(X.(D_i \,||\, id_i)) = (D_i \,||\, id_i)$.

6.6 Efficiency

The proposed scheme is attempted in that a member performs a constant amount of work in generating signature. As every file that needs to be signed is of fixed length, the group signatures of the proposed scheme satisfy the security property of unlinkability.

In each Sign operation, this proposed scheme can also be implemented by using the hash function H: $\{0,1\}^* \rightarrow \{0,1\}^k$ where k = 128 bits. Consequently, the signature of the message can be improved with the small size, and then the authentication of message can be obtained from this signature.

With a 1024 bit modulus, a proposed signature is about 1 Kbytes long. To be efficient with the time complexity, the proposed scheme can be supported as follows:

- In Sign operation, a group signature is required $2T_h + 1T_{exp} + 1T_N$ to generate, and
- The Outsider requires $2T_{exp} + 1T_N$ to verify the group signature.

In the above mentioned facts, some notations are used to analyze the computational complexity. These are:

- T_h is the time for executing the hash function $H(.)$.
- T_{exp} is the amount of time to execute a modular exponentiation operation.
- T_N is the time for multiplication with modulo n.

7. CONCLUSION

The signature scheme can be proposed an innovative application of group signature scheme for some organizations that wish to hide their internal organizational structures. This work proposes new prospects for the group signature scheme. In the proposed scheme, a group signature can be particularly generated with the collaboration of the group member and the issuing manager. As the procedures of this scheme, the storage space of group database can be saved because the proposed scheme does not need to store the lists of public keys of each member. Because of the control of issuing the signature from the authority of the issuing manager, the revoked members are not allowed to sign on behalf of group in future. In addition, every proposed signature can also give the authentication of the corresponding message by using hashing algorithm. The proposed scheme is attempted with better efficiency, and consequently, this signature scheme can be resulted to reduce the time complexity more than other previous scheme in Sign and Verify algorithm. As the future work, the group signature scheme will be implemented with stronger assumptions to be efficient for larger groups that are needed the dynamic revocation immediately. In addition, the group signature should be improved to apply among multi-groups efficiently.

8. REFERENCES

[1] M. S. H. Khiyal, A. Khan, S. Bashir, F. H. Khan and S. Aman. "Dynamic Blind Group Digital Signature Scheme in E-Banking", International Journal of Computer and Electrical Engineering, vol.3, No.4, pp. 514-519, 2011.

[2] X. Cheng, C. Yang and J. Yu. "A new approach to group signature schemes", Journal of Compters, vol.6, No.4, pp. 812-817, 2011.

[3] Petrank, E. and Kilian, J.: *Identity Escrow*, in Advances in Cryptology Crypto, (1998).

[4] Bellare, M., Shi, H. and Zhang, C.: "Foundations of Group Signatures: The Case of Dynamic Groups", In proceedings of CT-RSA '05, LNCS series, 3376, (2005) 136-153.

[5] Tsudik, G. and Ateniese, G.: "Quasi-efficient Revocation for Group Signatures", in To Appear in Financial Cryptography 2002, (2002).

[6] Camenisch, J.: *Efficient and Generalized Group Signatures*. In Proc. EURO-CRYPT 97, Springer-Verlag, Lecture Notes in Computer Science No. 1233, (1997) 465-479.

[7] J. Camenisch and M. Stadler. "Efficient group signature scheme for large groups", in Advances in Cryptology, Crypto '97, Lecture Notes in Computer Science, pp. 410-424, 1997.

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

Non-Functional Requirements Research: Survey

Harsimran Kaur,
ASU, Gurgaon

Dr. Ashish Sharma
GLA University, Mathura

Abstract— NFRs are important since the system architecture greatly depends on the NFRs [20]. Mostly NFR Literature has considered only for key challenges and issues related to NFR. In context of such a need a roadmap for important issues is required. In this paper survey has been presented on interesting ongoing work in the field of non functional requirements and tried to figure out the approaches and methods that are suggested in literature to deal with these issues.

Index Terms— Non Functional Requirements, Modeling, Identification, Formalization, Quantification, Automation

1. INTRODUCTION

IEEE Definition: "non functional requirement (NFR) – in software system engineering, a software requirement that describes not what the software will do, but how the software will do it, for example, software performance requirements, software external interface requirements, design constraints, and software quality attributes. Nonfunctional requirements are difficult to test; therefore, they are usually evaluated subjectively" [50]. In the past relatively little attention has been paid to the process of systematically dealing with NFR's and developers have relied mostly on their own intuitions, in an ad hoc way. In the years, the topic has attracted increasing interest from researchers, as testified by the many specialized events and workshops, as well as by the growing percentage of NFR papers in software engineering conferences.
There has been a considerable increase in the quantity of NFR research over the past few years (see Figure 1(b)). Despite the excellent work in the surveys listed earlier, there remains, to date, no comprehensive survey of the whole field of study concerning trends in research. This paper provides a range of options (Road map) for future research in area of non functional requirements.

The paper is organized as follows: Section 2 describes the result summary of the literature survey. Section 3 discusses the eight categories of research, and reviews the contributions from various research groups and the growing trend. Section 4 presents the conclusion and Future Work.

I. CLASSIFICATION SCHEME

The goal of our paper is to categorize the issues of NFR. We used the five digital libraries to search: ACM Digital Library, SpringerLink, ScienceDirect, Google Scholar, IEEE Xplore, ACM Digital Library, We classify these papers into eight categories.

1. Identification and Specification: Studies on notion(facet), classification and types of NFR.

2. Elicitation: Studies on requirements elicitation methods to empower requirement centered on NFRs.

3. Modeling (Informal): Studies on an approach to record and model non-functional requirements using UML and Relational Diagrams.

4. Modeling (formal): Studies on semantic concepts for the specification of non functional properties.

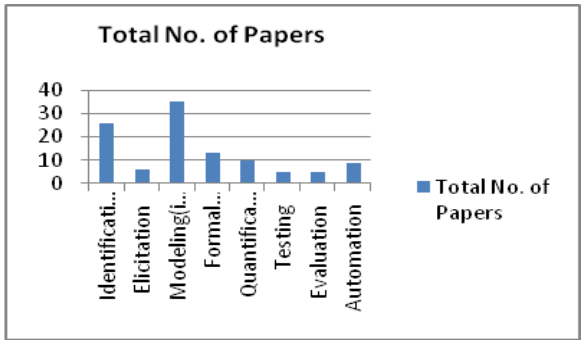

Fig 1(a) Paper in each category

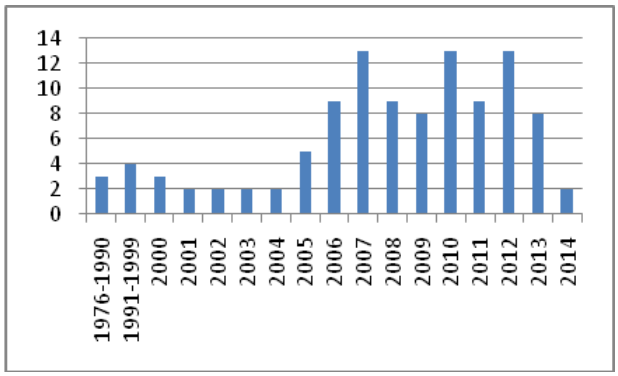

Fig 1(b) NFR Publication growth overtime

Quantification: Studies which explore number of avenues related to specification, design which deals and effect quantification of NFR.

5. Testing: Studies on issues, challenges while consider NFR, resulting from quality concern of stakeholders.

6. Automation: Studies on tools that assist the requirement Analyst while dealing with NFRs.

7. Evaluation: Studies on the degree to which NFR contributes to the improvement of software quality.

Assignment of category to each paper has been based on the main objective of the paper. Thus, in our classification some papers may be into another category by other researchers. For example many of the papers related to elicitation are presented by using any modeling language so we put them in the category of Modelling. Similarly identification and elicitation can be done by a single approach but we put them in different categories on the basis of focused concept used in the paper are Figure 1(a) shows the proportion of papers that fall into each of the different NFR area subject categories while Figure 1(b) shows the histogram charting NFR publication growth over time,

2. LITERATURE REVIEW

A. Identification and Specification

We surveyed different definitions and classification schemes proposed by different researchers. Critical Evaluation has been done as shown in Table 1. No Formal definition for NFR is found in literature except this: f: I → O (e.g., sum: int x int → int) which is defined by Chung etal [23]. NFR Framework is one of the prominent works that has been done in this field. For the specification of NFR there are three categories of approaches available in literature as shown in table 2. NFR Framework is the one of the important work which later on extended by number of researchers in order to solve the problem in their application domain [70]. Beside NFR Framework KAOS [103] and work with the help of UML has been found in literature which we are going to discussed in modelling. Formal language for NFR becomes a necessity but a familiar problem with formal methods in specifying such requirements is the high cost and difficulty of using them. Some of the work on formalization is shown in Table 2. There are few papers on the formal specification languages. Methods of supervised learning have been proposed in the literature to address the problem of identification and classification of NFR. Within the ECSS, ISO, and IEEE standards, a number of views and concepts are provided to describe various types of candidate portability requirements at the system, software, and hardware level [2].

B. Elicitation

There are only very few approaches and tools to elicit NFR. Many of the techniques and tools available are for functional requirements. Classification for the approaches used in elicitation of NFR as shown in Table 3 is based on different categories of approaches [108][116]. Ullah etal. has identified several key issues like conflicts of requirements, integration of NFR with FR and ambiguous specification of system features. They have found some of the solutions of these stated problems based on the available literature.

C. Modelling (informal)

A survey of the different works shows that most of them use UML with some extensions to add NFRs with the functional requirements models as shown in Table 4. UML proven to be successful modelling language to bring the revolution in NFR specification and modelling. Number of tools like Rational Rose, Smart Draw and Enterprise architect are available for UML.There are other approaches like Relational Model, Petrinets, Multimodel, NFR Framework, NFR Framework + that can be used to specify NFRs as mentioned in Table 4.

D. Formal Modelling

Formal methods offer a mathematical way to specify and analyze the behavior of NFR in a system together with a related tool support. Some relevant work done in this field by different researchers is discussed below. UML-B has been used for a real-time control system security concerns using an action systems approach [98]. There is requirement of tool support for UML-B. RoZ tool is used for modelling the airport security. It uses Z notation [63]. Another approach called KAMI is implemented as a distributed framework with a plugin architecture, which allows new tools to be incorporated to support other modeling notations and analysis procedures [34]. The approach is based on formal (probabilistic) models that are used at design time to reason about dependability of the application in quantitative terms. Another approach based on semantic concepts which form the basis of a semantic framework for the specification non-functional properties of component-based software [126][127]. Probabilistic way of characterizing the implementation of software non-functional requirements is proposed in [114]. SysML has been adopted as the modeling language by [107], since it enables requirement definition and can be formally extended. [90] has presented a semiformal approach for reasoning and refining functional requirements. Non Functional properties has expressed as NF-actions, NF-statements and NF-attribute. An another approach [83] aimed at lessen the risk of such misuses of quality models. It is centered on the definition of a language called *NoFun* which is to be used as a formal language for the exhaustive description of software quality. Borges and Mota [16] integrate UML class diagrams and *OhCircus* by written UML elements in terms of *OhCircus* constructs. *OhCircus* is a formal specification language which uses Z, CSP, calculus of Morgan and object-oriented theories. Casamayor [19, 20] propose a semi-supervised text categorization approach for the automatic identification and classification of non-functional requirements. Detection and classification of NFRs is performed using semi-supervised learning techniques. One more interesting work is shown in [26] which discuss an algebraic formalization of model based on graph theory which they use to prove safe termination in systems compliant with Ravenscar Computation Model (RCM), and show how to use the MAST+ static analyzer to verify the timing aspects. But till lot of work need to be done in this direction as mentioned in [19][21][23].

Table 1. Different Classification Schemes Proposed in Literature

Source	Research proposal	Critical Evaluation
Boehm(1976)[14]	Paper provides for the first time a clear, well-defined framework for assessing the often slippery issues associated with software quality, via the consistent and mutually supportive sets of definitions, distinctions and guidelines [14].	No emphasis has been found attributes of those NFRs.
McCall(1980)[72]	A Software Quality Measurement Manual was produced which contained procedures and guidelines for assisting software system developers in setting quality goals, applying metrics and making quality assessments.	It was assumed to be efficient model. Later it modified in 2000 where requirements are classified on the basis of product revision and transition.
Roman IEEE Computer (1985) [88]	It classifies requirements into interface, performance, operating, lifecycle, economic and political requirements.	It is complex classification.
Sommerville (1992) [100]	It considers organization, product and external aspects of requirement.	This model is accepted by many organizations but it could not sort the Non Functional Requirement specification issues.
Grady(1992)[42]	FURPS and FURPS+ is an acronym that represents the model. It introduces dimensions of quality.	Architectural integrity is not covered in the model
ISO/IEC 9126 (2001) [52]	Distinguishes four types of quality levels Quality in use, external quality, internal quality and process quality which helps to provide process oriented classification.	It sets standard for software practitioner to make the meaning of NFR and important NFR like performance clear to developers and users. But it is only limited to few NFRs.
Martin Glinz (2005) [39]	Presents New Classification of Requirements • Kind • Representation • Satisfaction • Role	Provides new notion to the NFRs but classification has no practical usefulness in daily life. It can be simplified further.
Jureta etal.(2006) [57]	This classification provides four categories: functional hardgoal, non functional hardgoal, functional softgoal and nonfunctional softgoal.	It is driven by nonfunctional perspective.
Martin Glinz (2007) [40]	Proposed New Definition to requirements and Specify classification rules based on Aspect-Oriented Representation.	Definition and Classification is less ambiguous than traditional definitions. Its Practical aspect needs to be find out.
Dewi Mairiza etal.(2010) [70]	It offers a novel classification of NFRs types based on types of systems and application domains.	It presents comprehensive lists of NFR types which helps developer to identify NFR for their particular system. But the Terminology present does not improve the notion of NFRs.
Chi-Lun Liu (2010) [66]	Proposes top level NFR ontology helpful in conflict detection between NFRs which is extended from Glinz's study.	Nothing has been done for improving NFR facet in it.

Table 2: Categorization of Approaches used in Identification

	[2]	[18]	[19]	[20]	[22]	[23]	[25]	[28]	[29]	[34]	[35]	[37]	[39]	[40]	[42]	[48]	[65]	[66]	[70]	[80]	[125]	[126]
Informal Approach(I)	I				I	I	I	I	I				I	I	I	I			I	I		
Semi-Formal Approach(S)			S	S													S	S				
Formal Approach(F)		F								F	F	F									F	F

Table 3: Classification of approaches used in elicitation of NFRs

Approaches	References	Description
NFR Framework Based	[4][27][48][84][101]	Process-oriented and qualitative method for handling NFRs
Quality Model Based	[6][9][52][58][104]	NFR method consists of quality attributes, based on quality model
NFR Framework with Quality Model Based	[52][56][58][121]	i* framework and meta-model are presented in these approach.
Guideline Based (Without NFR Framework and Quality Model Based	[51][111]	Approaches set the and focused on gathering only the minimum set of information on quality goals.

Table 4: Informal Modelling of NFR

Approach(Model)	Purposed
Use Case and Goal Driven (2005)[103]	Integrates FR with NFR at design level by using use case elements.
Extended Use Case (2006)[11]	To separate(cross-cut) the concerns at the requirements level(on tha basis of application domain) that can be achieved by checking concerns that produce spread and tangled representation that are difficult to understand and maintain. Extended elements helps to express and integrate NFR and challenges to requirement analyst with the FR.
OONFR(2001) [26][27]	UML Class Diagram is proposed which use LEL of UofD as input and class diagram has signals of what elements (classes, attributes, operations and relationship) are responsible for NFR.
Extended UML(2007)[123]	Dependency notation is introduced in UML to model design decisions.
Extending UML with NFR Framework(2005)[22][23]	Meta-Model to represent concepts in NFR Framework and made extension in UML and NFR Framework to integrate the notations of two modeling languages.
Novel Framework with UML(2005)[106]	UML design is integrated to NFRs for the purpose of reengineering process of legacy systems.
Layered Model(Conceptual ArchitecturalModel)(2005)[117]	Additional layer is added to traditional architectural model for satisfying NFR Role.
UMLsec (2002) [53]	UML extension mechanism based on formal semantics to evaluate security aspects of system design
SecureUML (2002) [68]	UMl extension mechanism to specify information for access control in the design of application.
Abuse Case Model(2002)[68]	Extended Use Case Model to capture and analyze security requirements by specifying check on interaction between system and actors.
Architectural pattern(2005) [61]	Improving system dependability and trustworthiness by improving the modeling of NFR(operationalizable NFRs, and checkable NFRs)
NFR Framework with Role Activity Model (2007)[3]	Remodelling business process to better representation and realization of NFR aspects of processes by linking RAD with NFR graphic facility.
Use Case extended to Control Case(2006)[114][124]	Focus is made on operating conditions by adding control cases to 4+2 view of architecture(UML Process view)
DERAF(2007) [69]	Combines the use of aspects with RT-UML, aiming to separate the handling of non-functional from functional requirements in the Model Driven Design of DERTS.
Integrated Model(2007) [113]	Approach is based on building a base quality model that relies on an explicit meta-model. Purpose models are also needed to support the planning and realization of quality assurance are derived from the base model by quantifying the relations modeled in the base model.
XML-NFR(2007) [117]	It is also step to integrate with functional requirements design model based on

	simple language XML but not solved many issues of NFR.
Pluggable Framework (Wireless Sensor networks)(2008) [15]	It allows WSN applications in TinyDDS(Data Distributed Services) to have fine-grained control over non-functional properties and specialize in their own requirements.
RASF(2008) [45]	A Multi-Agent Systems (MAS) approach in Reactive Autonomic Systems (RAS) whose specifications are mentioned in single formal framework.
XML with Petri-Nets(2008)[31]	The intermediate model is based on XML and indicates the relationship between the entities of design models and analysis models by minimizing the gaps
Extended PLUS (UML Based Model) (2009) [79]	It provide a unified and systematic framework for analysis modeling of NFRs in Software Product Lines by integrating it with Lines PLUS.
SysML(2009)[107]	It represents NFR as how non-functional requirements are related between them and to system components forming the overall system architecture.
KAMI(2009) [38][103]	Approach relies on run-time monitoring and uses the data collected by the probes to detect if the behavior of the open environment in which the application is situated can lead to a failure of the application
NFR with AORE(2010)[114]	It map non-functional requirements into function and architectures through non-functional scenario template which improves traceability from requirement analysis level to implement level.
Configuration Models(2012) [33]	Mapping nonfunctional aspects to given commercial-off-the-shelf modules which makes possible the integration of commercial software modules into product families
Relational Model (2011) [59]	It introduce change management mechanism that trace the the impact of NFRs on the other constructs in the ontology such as FR or NFR operationalization and vice versa.
Multimodel Approach(2012) [41]	Besides the refinement of NFR it allows the validation of its fulfillment through the application of metrics that are associated to each NFR.
Design Patterns Approach(2001) [44]	It provides guidance and reasoning support when applying patterns during the design of a software system.
UML with OCL [83]	Non functionality is described by means of a notation called NoFun, which allows us to introduce non-functional attributes of software
UML Based on EAST-ADL (2011) [91]	Provides solution for telecommunication systems for modeling product families, targeting cost sensitivity non-functional requirements and performing cost analysis.
Domain Specific Modelling Approach(NFR+ Framework) [119]	The solution enables a full bi-directional traceability from the requirements to models to the implementation.
UML for intrusion specification(2006) [49]	UML notations extended to suit the context of intrusion scenarios that allows developers to specify intrusions

E. Quanitification

There are very few languages to state non-functionality in form so that it can be quantified. One of the language [35][83] is NoFun which provides a common framework in which people can formulate, analyse and compare their proposals about non-functionality. A measure for reusability is refined by this language.The combination of both NoFun and the implementation selection algorithm can be an aid to software specification, design, reusability and maintenance. Stephan Jacobs [54] of Ericsson presented a case study on improving requirement engineering. From the concepts offered in Planguage Jacobs proposed that Gist, Scale, Meter, Past, Record, Must, Plan and Wish should be made visible in our requirements specifications by using keywords in bold letters. GIST is a rough summary of the requirement. According to Jacobs SCALE defines the unit in which the requirements has to be measured. METER defines the way how the measurement will be performed. PAST and RECORD are benchmarks. Past is a value which is typical for (own) products developed in the past. MUST, PLAN and WISH envisage the future. Must, Plan and Wish characterize the system that is to be built. Affleck [5] extends the previous quantitative reasoning extension into a single objective optimization model that aims to selectively choose operationalizations in order to increase the overall satisfaction of non-functional requirements. One metric is proposed in [1]

that can be used in the early stages of software development projects to estimate effort of new projects. Affleck [4] presents a process-orientated, lightweight, quantitative extension to the NFR Framework; focusing on providing quantitative support to the decision process and how decisions affect the system. Some key issues related to NFR quantification are discussed in [85][90].They discussed issues related to sharing of information between customer and supplier as it is must for optimal quantification. Requirements Convergence Plan can be used to create better NFR quantification circumstances for customers and suppliers. Another evaluation model of NFR is proposed in [94] which mainly focusing on the user maintenance and operation issues. This model consists of NFR categories, NFR metrics, description level grading and weight to each NFR. Another contribution to quantification of NFR is made by Bin [119] by proposing three methods for calculating non-functional properties. The cumulative method is applied to calculation of energy cost, memory cost, and number of defects and so on. The multiplicative method is applied to calculation of non-functional properties which can be described by probability, such as reliability, confidentiality. The graphic method is applied to calculation of consumed time. Paper also proposes a 0-1 programming method for selecting the best non-functional requirement implementation strategies. Bhatti etal.[13] tries to quality metrics on the basis of UML diagrams.

F. *Testing*

[97] has mentioned prevalent testing issues in the light of NFR. There is a great need to work on specification for testability, design for testability and code for testability as mentioned in previous sections of paper. They also mention certain research direction for future exploration in their paper. One of the solutions is to have aspect-oriented techniques. It offers a promising approach for capturing such issues under verification. In the Literature we found very few testing techniques (NFR) proposed that too are application based as mentioned in Table [5].

G. *Automation*

Automation of process is necessary for speed up the development process. Not only requirements have to be carefully considered but they also have to be implemented. But there need to validate the implementation which can be easily done with the help of tool. Tool can potentially help agile software development teams in reasoning about and visually modeling NFRs as first-class artifacts early on during requirements gathering and analysis phases. It is better to create a simple and open toolkit that in turn can be adapted to a variety of projects and architectures [30]. Since there is need to handle NFRs automatically various tools have been proposed by researchers as enlisted in Table 6.

H. *Evaluation*

Evaluation means NFR importance degree assessment given by the expert's team on the basis of certain variables. Some important evaluation has been done in [12][81][105] [123]

3. CONCLUSION AND FUTURE SCOPE

NFR needs to look after starting from the early stages of software development. There is a need to chance the facet of NFRs while specifying it in SRS. Most of the literature is based on NFR elicitation and NFR Framework (informal approach). Some formal approach needs to be work out. According to Singh et al. number of models are available for Functional Requirements like Four Variable Model, COCOMO. Model and Reference Model etc. but no standardized model has been found opted for NFR [21][97]. It is due to informal presentation, NFR still a challenge in the field of requirement engineering.

To complete the specification for NFRs besides the four variables (NAT, REQ, IN, OUT) of four variable model new variable can be introduced or new model can be introduced from scratch for dealing with NFRs. Extensions to this model is also suggested in [77]. Similarly Reference Model can be extended for NFR as mentioned by Chung.[21].

As proposed in [97] NFR can be handled more concretely by MBT, some approach needs to be work out for handling real life situations.

Aspects help to achieve modularity in software development process. The use of AO to deal with NFR has already been proposed in [69].

In this paper we surveyed different aspects of NFR. We are likely to focus on formal modeling of NFR in future work. As it is the foremost challenge that need to be overcome if we want NFR to be quantified. Some of the issues related to NFR are mentioned in Table 7 found from the literature survey.

Table 5. Testing issues based on application

Application	Issues	Solution Proposed
Web Based application[89]	Verification during testing	• Metrics for the navigability • Load and Performance Testing
Quality verification of mobile phones [76]	Lack of Tool Support Classify types of NFR	Aspect-oriented techniques

TABLE 6: TOOLS

Source	Tool or Approach Name	Purpose
Jan Ladiges etal.(2013) [62]	Presented a set of non-functional requirements on automated production facilities.	Approach used detects unintentional changes in its behavior after performing modifications.
Farid and Mitropoulos(2012)[33]	NORMATIC	Modeling for Agile Processes (NORMAP) Methodology.
Cesare etal.(2012) [84]	Q4BPMN	Non-Functional requirements can be directly expressed within the BPMN model.
Janne Merilinna etal.(2012)[75]	Supported by a tool enables to do that in the context of Domain-Specific Modeling (DSM).	Bi-directional traceability link between requirements and implementation is maintained by NFR+ Framework.
Kristoffer Dyrkorn(2008)[30]	Present an open-source toolkit that enables automated testing of non-functional requirements.	Provides developers and project managers with reports about the system under development.
Jane Cleland etal.(2007)[24]	Technique for automating the detection and classification of non-functional Requirements.	Approach is used to detect and classify stakeholders' quality concerns across requirements specifications containing scattered and non-categorized requirements
Lawrence Chung etal.(1996) [22]	Address tool support for the change process	Approach is based on existing NFR Framework
Al Balushi etal.(2007)[6]	ELICITO	Quality ontology-guided NFR elicitation tool

TABLE 7: NFR ISSUES

S.No	Categories	Issues
1	Identification and Specification	• Need to change the notion while specifying NFR in RFP [39][40] • Formal Techniques [20][21]
2	Elicitation	• Aspect oriented documentation of attributes and constraints [69][108]
3	Formal Modeling	Need to develop formal models like • Four Variable Model[21][77] • Reference model[21]
4	Quantification	• Formal models helps in quantification of NFR [77][81][93]
5	Testing	• Clear Identification and Specification(Formal techniques) • AOP[69][97] • MBT[41][97]
6.	Automated Tool	• Tool for empowering analyst by providing knowledge repository, detecting NFR conflict and to assess impact of NFRs in early stages [5][30].

4. REFERENCES

[1] Abdukalykov, R.; Hussain, I.; Kassab, M.; Ormandjieva, O., "Quantifying the Impact of Different Non-functional Requirements and Problem Domains on Software Effort Estimation," Software Engineering Research, Management and Applications (SERA), 2011 9th International Conference on , vol., no., pp.158,165, 10-12 Aug. 2011

[2] Abran A. , k. T. Al-Sarayreh, and J. J. Cuadrado-Gallego, 'A Standards-based Reference Framework for System Portability Requirements', Computer Standards and Interfaces Journal, Elsevier, Vol 35, 2013, pp. 380-395.

[3] Aburub, F., Odeh, M., Beeson, I.: Modelling non-functional requirements of business processes. Information and Software Technology 49, 1162–1171 (2007)

[4] Affleck A., Krishna, A. and Achuthan, N.R. (2013). Optimal Selection of Operationalizations for Non-Functional Requirements. In Proc. Conceptual Modelling 2013 (APCCM 2013) Adelaide, Australia. CRPIT, 143. Ferrarotti, F. and Grossmann, G. Eds., ACS. 69-78

[5] Affleck, A.; Krishna, A., "Supporting quantitative reasoning of non-functional requirements: A process-oriented approach," Software and System Process (ICSSP), 2012 International Conference on , vol., no., pp.88,92, 2-3 June 2012.

[6] Al Balushi T. H., P. R. F. Sampaio, D. Dabhi, and P. Loucopoulos, "ElicitO: a quality ontology-guided NFR elicitation tool," in Proceedings of the 13th international working conference on Requirements engineering: foundation for software quality (REFSQ'07), Springer-Verlag, Berlin, Heidelberg, pp. 306-319, 2007.

[7] Allen H. Dutoita and Barbara Paechb Rationale-Based Use Case Specification Requirements Eng (2002) 7:3–19 2002 Springer-Verlag London Limited

[8] Ameller, D.; Ayala, C.; Cabot, J.; Franch, X., "How do software architects consider non-functional requirements: An exploratory study," Requirements Engineering Conference (RE), 2012 20th IEEE International , vol., no., pp.41,50, 24-28 Sept. 2012

[9] Anandha G.S. Mala and G.V. Uma, "Elicitation of non functional requirement preference for actors of Usecase from domain model," PKAW 2006, LNAI 4303, Springer-Verlag Berlin Heidelberg, pp. 238-243, 2006.

[10] Ben A. Hamida, A. Bertolino, A. Calabrò, G. De Angelis, N. Lago, and J. Lesbegueries, "Monitoring service choreographies from multiple sources, " in Proc. of 4th International Workshop on Software Engineering for Resilient Systems (SERENE 2012), ser. LNCS, vol. 7527. Springer, Sept. 2012, pp. 134-149.

[11] Berenbach Brian and Mark Gall.(2006) Toward a Unified Model for Requirements Engineering. ICGSE, 0:237–238.

[12] Berntsson Svensson, R., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., Feldt, R., Aurum, A.: Prioritization of quality requirements state of practice in eleven companies. In: RE 2011, pp. 69–78. IEEE (2011)

[13] Bhatti Shahid Nazir, ACM (Software Engineering Notes) Journal, "Deducing the complexity to quality of a system using UML", ISSN: 0163-5948, Volume 34, Pages 1-7, USA.

[14] Boehm B. et al. (1976). Quantitative Evaluation of Software Quality. Proceedings of the 2nd IEEE International Conference on Software Engineering. 592-605.

[15] Boonma, P.; Suzuki, J., "Middleware Support for Pluggable Non-Functional Properties in Wireless Sensor Networks," Services - Part I, 2008. IEEE Congress on , vol., no., pp.360,367, 6-11 July 2008

[16] Borges R. and A. Mota, Integrating UML and Formal Methods, Electronic Notes in Theoretical Computer Science, 184, pp. 97-112, 2003

[17] Bret Pettichord. (October 2002), Design for Testability ,Pacific Northwest Software Quality Conference, Portland, Oregon.

[18] Cancila D. , R. Passerone , T. Vardanega and M. Panunzio "Toward correctness in the specification and handling of non-functional attributes of high-integrity real-time embedded systems", IEEE Trans. Ind. Informat., vol. 6, no. 2, pp.181 - 194 2010

[19] Casamayor A, Godoy D, Campo M (2009) Semi-supervised classification of non-functional requirements: an empirical analysis. Rev Iberoam Intel Artif 13(44): 35–45. doi:10.4114/ia.v13i44.1044

[20] Casamayor A, Godoy D, Campo M (2010) Identification of non-functional requirements in textual specifications: a semi-supervised learning approach. Inform Softw Technol 52(4): 436–445. doi:10.1016/j.infsof.2009.10.010

[21] Chung L. , Kendra Cooper , Anna Yi, Developing adaptable software architectures using design patterns: an NFR approach, Computer Standards & Interfaces, v.25 n.3, p.253-260, June 2003

[22] Chung L. a, Brian A. Nixon b and Eric Yu c Dealing with Change: An Approach Using Non-functional Requirements 1 Requirements Eng (1996) 1:238-260-9 1996 Springer-Verlag London Limited

[23] Chung, L., Nixon, B., Yu, E. and Mylopoulos,J. (2000) "Non-Functional Requirements in Software Engineering": Boston Kluwer Academic Publishers.

[24] Cleland-Huang Jane, Raffaella Settimi, Xuchang Zou , Peter Solc Automated classification of non-functional requirements Requirements Eng (2007) 12:103–120 Springer-Verlag London Limited 2007

[25] Curtis B. in blog for Consortium for IT Software Quality http://it-cisq.org/non-functional-requirements-be-here/

[26] Cysneiros Luiz Marcio and Julio Cesar Sampaio Leite (2001). Using UML to reflect Non-functional Requirements. In CASCON, page 2.

[27] Cysneiros Luiz Marcio, Member,Julio Cesar Sampaio do Prado Leite, Member,(MAY 2004) Nonfunctional Requirements:From Elicitation to Conceptual Models, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 30, NO. 5.

[28] do Prado Sampaio.T. Borgida et al. (2009). On Non-Functional Requirements in Software Engineering (Eds.): Mylopoulos Festschrift, LNCS 5600, pp. 363–379, 2009.© Springer-Verlag Berlin Heidelberg.

[29] Dobson G., S. Hall, and G. Kotonya. A domain-independent ontology for non-functional requirements. In ICEBE '07: Proceedings of the IEEE International Conference on e-Business Engineering, pages 563-566, Washington, DC, USA, 2007. IEEE Computer Society.

[30] Dyrkorn K. and F. Wathne, "Automated testing of non-functional requirements," in Companion to the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications, New York, NY, USA, 2008.

[31] Emadi, S.; Shams, F., "An approach to non-functional requirements analysis at software architecture level," Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on , vol., no., pp.736,741, 8-11 July 2008

[32] Fabbrini F., M.Fusani, S.Gnesi, G.Lami "Quality Evaluation of Software Requirement Specifications", Proc of Software & Internet Quality Week 2000 Conference., San Francisco, CA May 31-June 2 2000, Session 8A2.

[33] Farid W.M. and F. J. Mitropoulos, "Novel lightweight engineering artifacts for modeling non-functional requirements in agile processes", Proc. IEEE SoutheastCon 2012 (SoutheastCon 2012), Mar. 2012.
[34] Filieri A., C. Ghezzi, and G. Tamburrelli, "A formal approach to adaptive software: continuous assurance of non-functional requirements," Formal Aspects of Computing, vol. 24, no. 2, pp. 163-186, 2012.
[35] Franch X. "Systematic Formulation of Non-functional Characteristics of Software," Proc. 3rd IEEE Int',l Conf. Requirements Eng. (ICRE), IEEE CS Press, Los Alamitos, Calif., 1998, pp. 174-181.
[36] Friess, Wolfgang; Kubica, Stefan, "Implementing Non-Functional Requirements (29 March 2006) Â¿ A Layered Modeling Approach," Model-Based Testing, ITGA FA 6.2 Workshop on and GI/ITG Workshop on Non-Functional Properties of Embedded Systems, 2006 13th GI/ITG Conference -Measuring, Modelling and Evaluation of Computer and Communication (MMB Workshop) , vol., no., pp.1,7, 27-
[37] Galster M. and E. Bucherer, "A Taxonomy for Identifying and Specifying Non-Functional Requirements in Service-Oriented Development," IEEE Congress on Services, IEEE Press, 2008, pp. 345-352.
[38] Ghezzi C.and G. Tamburrelli. Reasoning on non-functional requirements for integrated services. In RE'09, Atlanta, USA, 2009.
[39] Glinz M., "Rethinking the notion of non-functional requirements," in Third World Congress for Software Quality, Munich, Germany, 2005, pp. 55-64.
[40] Glinz, M. "On non-functional requirements," in 15th IEEE International Requirements Engineering Conference (RE '07), 2007, pp. 21-26.
[41] Gonzalez-Huerta, J., Insfran, E., Abrahao, S., McGregor, J.D.: Non-functional Requirements in Model-Driven Software Product Line Engineering. In: 4th Int. Workshop on Non-functional System Properties in Domain Specific Modeling Languages, Insbruck, Austria (2012)
[42] Grady R and Caswell D, Software Metrics: Establishing a Company-wide Program, Prentice Hall, Englewood Cliffs, New Jersey, 1987
[43] Grimshaw D.J., W. Godfrey and G.W. Draper, "Non-Functional Requirements Analysis: Deficiencies in Structured Methods," Information &, Software Technology, vol. 43, no. 11, pp. 629-635, 2001.
[44] Gross Daniel and Eric Yu From Non-Functional Requirements to Design through Patterns Requirements Eng (2001) 6:18–36 Springer-Verlag London Limited
[45] Heng Kuang; Ormandjieva, O., "Self-Monitoring of Non-functional Requirements in Reactive Autonomic Systems Framework: A Multi-agent Systems Approach," Computing in the Global Information Technology, 2008. ICCGI '08. The Third International Multi-Conference on , vol., no., pp.186,192, July 27 2008-Aug. 1 2008
[46] Hosono S., T. Hara, Y. Shimomura and T. Arai, Prioritizing Service Functions with Non-Functional Requirements, CIRP Industrial Product-Service Systems Conf., 2010, pp.133-140.
[47] http://en.wikipedia.org/
[48] Huang J.C., R. Settimi, X. Zou, and P. Solc, "The detection and classification of non-functional requirements with application to early aspects," Center for Requirements Engineering, School of Computer Science, Telecommunications and Information Systems, DePaul Üniversity, In 14th IEEE International Requirements Engineering Conference (RE´06), 2006.
[49] Hussein, M.; Zulkernine, M., "UMLintr: a UML profile for specifying intrusions," Engineering of Computer Based Systems, 2006. ECBS 2006. 13th Annual IEEE International Symposium and Workshop on , vol., no., pp.8 pp.,288, 27-30 March 2006
[50] IEEE (1993). IEEE Recommended Practice for Software Requirements Specifications.IEEE Standard 830-1993.
[51] In H., and B. W. Boehm, "Using win-win quality requirements management tools: a case study", Annals Software Eng. 11(1): pp. 141-174, 2001.
[52] ISO/IEC 9126-1:2001(E): Software Engineering - Product Quality - Part 1: Quality Model(2001)
[53] J¨urjens Jan.(2002) UMLsec: Extending UML for Secure Systems Development. In UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language, pages 412–425, London, UK, Springer-Verlag.
[54] Jacobs S., "Introducing Measurable Quality Requirements: A Case Study," Proc. Fourth Int',l Symp. Requirements Eng., pp. 172-179, 1999.
[55] Jane Cleland-Huang, Toward improved traceability of non-functional requirements, Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering, November 08-08, 2005, Long Beach, California
[56] Jaramillo A.F., "Non-functional requirements elicitation from business process models", IEEE Doctoral Paper 2011.
[57] Jureta, I.J., Faulkner, S., Schobbens, P.-Y.: A more expressive softgoal conceptualization for quality requirements analysis. In: Embley, D.W., Olivé, A., Ram, S. (eds.) ER 2006. LNCS, vol. 4215, pp. 281–295. Springer, Heidelberg (2006)
[58] Kaiya H., A. Osada, and K. Kaijiri, "Identifying stakeholders and their preferences about NFR by comparing use case diagrams of several existing systems," IEICE - Trans. Inf. Syst. E91-D, pp. 897-906, April 2008.
[59] Kassab, M.; Ormandjieva, O.; Daneva, M., "Relational-model based change management for non-functional requirements: Approach and experiment," Research Challenges in Information Science (RCIS), 2011 Fifth International Conference on , vol., no., pp.1,9, 19-21 May 2011
[60] Kavakli E. and P. Loucopoulos, "Goal Modeling in Requirements Engineering: Analysis and Critique of Current Methods", Information Modeling Methods and Methodologies: Advanced Topics of Database Research, IGI Publishing, 2005, pp. 102-124.
[61] L. Xu, H. Ziv, and T. A. Alspaugh, et. al., "An architectureal pattern for non- functional dependability requirements". Systems & Software, Vol.79, No. 10, pp.1370-1378.
[62] Ladiges, J.; Fay, A.; Haubeck, C.; Lamersdorf, W., "Operationalized definitions of non-functional requirements on automated production facilities to measure evolution effects with an automation system," Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on , vol., no., pp.1,6, 10-13 Sept. 2013
[63] Ledru Y., R. Laleau, M. Lemoine, S. Vignes, D. Bert, V. Donzeau-Gouge, C. Dubois, and F. Peureux. An attempt to combine UML and formal methods to model airport security. In Forum of the 18th International Conference on Advanced Information Systems Engineering, pages 47-50, Luxembourg, 2006.
[64] Lera, I.; Puigjaner, R.; Semantic Layered Architecture to integrate FR/NFR in Software Performance Engineering. WOSP/SIPEW 2010, poster
[65] Liu Xiaoqing; Azmoodeh, M.; Georgalas, N., "Specification of non-functional requirements for contract specification in the NGOSS framework for quality management and product

evaluation,"Software Quality, 2007. WoSQ'07: ICSE Workshops 2007. Fifth International Workshop on , vol., no., pp.,, 20-26 May 2007

[66] Liu, (2010) "Ontology-Based Conflict Analysis Method in Non-Functional Requirements" 978-0-7695-4147-1/10 IEEE.

[67] Livson B.U., A practical approach to software quality assurance, ACM SIGSOFT Software Engineering Notes, v.13 n.3, p.45-48, July 1988

[68] Lodderstedt Torsten, David A. Basin, and J¨urgen Doser.(2002) SecureUML: A UMLBased Modeling Language for Model-Driven Security. In UML '02: Proceedings of the 5th International Conference on The Unified Modeling Language, pages 426–441, London, UK, 2002. Springer-Verlag.

[69] M.A. Wehrmeister, E. P. Freitas, C. E. Pereira, and F. Wagner, "An Aspect-Oriented Approach for Dealing with Non-Functional Requirements in Model-Driven Development of Distributed Embedded Real-Time Systems", 10th IEEE Symposium on Object Oriented Real-Time Distributed Computing, 2007.

[70] Mairiza D.et al. (2010) , An investigation into the notion of non-functional requirements, Proceedings of SAC'10, Pages 311-317, ACM New York, NY, USA

[71] Matoussi, R. Laleau. (October 2008) A Survey of Non-Functional Requirements in Software Development Proces, Technical Report TR–LACL.

[72] McCall, J.A., Matsumoto, M.T. (1980). Software Quality Measurement Manual, Vol. II. Rome Air Development Center, RADC-TR-80-109-Vol-2.

[73] Mehta, R. and Chung, L. "Dependencies among Architectural Components and their Impacts towards Non-Functional Requirements," IEEE Software, Special Issue on Twin Peaks of Requirements and Architecture(2013).

[74] Mens, Tom; Wermelinger, Michel; Ducasse, Stane; Demeyer, Serge; Hirschfeld, Robert and Jazayeri, Mehdi (2005). Challenges in Software Evolution. In 8th conference on on Principles of**Software Evolution**, p.13-22, September 05-06, **2005**

[75] Merilinna Janne etal. NFR+ framework method to support bi-directional traceability of non-functional requirements Springer-Verlag 2012

[76] Metsa, J.; Katara, M.; Mikkonen, T., "Testing Non-Functional Requirements with Aspects: An Industrial Case Study," Quality Software, 2007. QSIC '07. Seventh International Conference on , vol., no., pp.5,14, 11-12 Oct. 2007

[77] Miller, S.P. and Tribble, A.C. 2001. Extending the Four-Variable Model to Bridge the System-Software Gap. In Proceedings of the Twentith IEEE/AIAA Digital Avionics Systems Conference (DASC'01).

[78] Mylopoulos J., L. Chung, B. Nixon (1992). Representing and Using Nonfunctional Requirements: A Process- Oriented Approach. IEEE Transactions on Software Engineering 18, 6 (June 1992). 483-497.

[79] Nguyen, Q.L., "Non-functional requirements analysis modeling for software product lines," Modeling in Software Engineering, 2009. MISE '09. ICSE Workshop on , vol., no., pp.56,61, 17-18 May 2009

[80] Norian, M., Bagheri, E., Du, W.: Non-functional Properties in Software Product Lines: A Taxonomy for classification. In proc. of 24th International Conference on Software Engineering and Knowledge Engineering (SEKE 2012), 2012.

[81] Okubo, T.; Yoshioka, N.; Kaiya, H., "Security Driven Requirements Refinement and Exploration of Architecture with Multiple NFR Points of View," High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium on , vol., no., pp.201,205, 9-11 Jan. 2014

[82] Osterweil, L.J., "A Future for Software Engineering?," Future of Software Engineering, 2007. FOSE '07 , vol., no., pp.1,11, 23-25 May 2007.

[83] P. Botella, X. Burgués, X. Franch, M. Huerta, G. Salazar. "Modelling Non-Functional Requirements". Proceedings of Jornadas Ingeniería de Requisitos Aplicados (JIRA), 2001

[84] Pavlovski C.J. J. and Zou, "Non-Functional Requirements in Business Process Modeling", Proc. of 5th Asia-Pacific Conference on Conceptual Modelling, Wollongong, NSW, Australia. CRPIT, 2008, Vol. 79, pp. 103-112.

[85] Poort, E.R., Key, A., de With, P.H.N., van Vliet, H.: Issues Dealing with Non-Functional Requirements across the Contractual Divide. In: WICSA/ECSA 2012, pp. 315–319 (2012).

[86] Pressman, R. S. (2005). Software Engineering: A Practitioner's Approach (6th ed.). New York:- McGraw-Hill Publication.

[87] Rao Ananda and M.Gopichand (2011)" Four Layered Approach to non-functional requirements Analysis" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 2.

[88] Roman, G.-C.: A TaxonoTaxonomy of Current Issues in Requirements Engineering. IEEE Computer, 14–21 (April 1985)

[89] Romano Breno, Gláucia Braga e Silva, Henrique Fernandes de Campos, Ricardo Godoi Vieira, Adilson Marques da Cunha, Fábio Fagundes Silveira, Alexandre Carlos Brandão Ramos, "Software Testing for Web-Applications Non-Functional Requirements," itng, pp.1674-1675, 2009 Sixth International Conference on Information Technology: New Generations, 2009

[90] Rosa N., P. Freire Cunha, and G. Ribeiro Justo. An Approach for Reasoning and Refining Non-Functional Requirements. Journal of the Brazilian Computer Society, 10:62-84, 2004. 10.1007/BF03192354.

[91] Saadatmand, M., Cicchetti, A., Sjödin, M.: Uml-based modeling of non-functional requirements in telecommunication systems. In: The Sixth International Conference on Software Engineering Advances, ICSEA 2011 (2011)

[92] Sadana, V.; Liu, X.F., "Analysis of Conflicts among Non-Functional Requirements Using Integrated Analysis of Functional and Non-Functional Requirements," Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International , vol.1, no., pp.215,218, 24-27 July 2007

[93] Sadiq, J.; Mohsin, A.; Arif, F., "Quantifying Non-functional Requirements in Service Oriented Development," Frontiers of Information Technology (FIT), 2011 , vol., no., pp.224,229, 19-21 Dec. 2011 doi: 10.1109/FIT.2011.48

[94] Saito Yasuhiro," Evaluation of Non Functional Requirements in a Request For Proposal (2012)"in Seventh International Conference on Software Process and Product Measurement, 978-0-7695-4840-1/12 IEEE

[95] Salazar-Zárate G., P. Botella. "Use of UML for Non-Functional Aspects". Proceedings of 13th International Conference Software & Systems Engineering and their Applications (ICSSEA), 2000.

[96] Schmeling, B.; Charfi, A.; Thome, R.; Mezini, M., "Composing Non-functional Concerns in Web Services," Web Services (ECOWS), 2011 Ninth IEEE European Conference on , vol., no., pp.73,80, 14-16 Sept. 2011

[97] Singh Pratima and Anil Kumar Tripathi, (July 2012) "ISSUES IN TESTING OF SOFTWARE WITH NFR", International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.4.

[98] Snook Colin , Michael Butler, UML-B: Formal modeling and design aided by UML, ACM Transactions on Software Engineering and Methodology (TOSEM), v.15 n.1, p.92-122, January 2006

[99] Soltani Samaneh , Mohsen Asadi , Dragan Gašević , Marek Hatala , Ebrahim Bagheri, Automated planning for feature model configuration based on functional and non-functional requirements, Proceedings of the 16th International Software Product Line Conference, September 02-07, 2012, Salvador, Brazil

[100] Sommerville (2006).6th edition Ed. Addison Wesley,

[101] Song Xiping; Hwong, B.; Ros, J., "Lessons from Developing Nonfunctional Requirements for a Software Platform," Software, IEEE , vol.29, no.2, pp.74,80, March-April 2012

[102] Subramanian Nary , Lawrence Chung, Software architecture adaptability: an NFR approach, Proceedings of the 4th International Workshop on Principles of Software Evolution, September 10-11, 2001, Vienna, Austria

[103] Supakkul Sam and Lawrence Chung.(2004) Integrating FRs and NFRs: A Use Case and Goal Driven Approach. Proc. SERA 04, pages 30–37,.

[104] Taeg Jung Keo; Gil-Haeng Lee, "A systematic software development process for non-functional requirements," Information and Communication Technology Convergence (ICTC), 2010 International Conference on , vol., no., pp.431,436, 17-19 Nov. 2010

[105] Thakurta R (2012) A framework for prioritization of quality requirements for inclusion in a software project. Software quality journal. Springer, New York

[106] Tonu Subrina A. and Ladan Tahvildari. Towards a Framework to IncorporateIn Proceedings of IEEE WCRE Workshop on Reverse Engineering to Requirements (RETR), Pittsburgh, Pennsylvania, USA, pages 13–18, November 2005.

[107] Tsadimas, A.; Nikolaidou, Mara; Anagnostopoulos, D., "Handling Non-functional Requirements in Information System Architecture Design," Software Engineering Advances, 2009. ICSEA '09. Fourth International Conference on , vol., no., pp.59,64, 20-25 Sept. 2009

[108] Ullah Saeed, Muzaffar Iq bal, Aamir Mehmood Khan,(2011) A Survey on Issues in Non-Functional Requirements Elicitation Computer Networks and Information Technology (ICCNIT), 2011 International Conference on 978-1-61284-941-6/11 IEEE.

[109] Umar, M.; Khan, N.A., "Analyzing Non-Functional Requirements (NFRs) for software development," Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on , vol., no., pp.675,678, 15-17 July 2011

[110] Van Lamsweerde Axel. Goal-Oriented Requirements Engineering: A Guided Tour. In RE ’01: Proceedings of the Fifth IEEE International Symposium on Requirements Engineering (RE ’01), page 249, Washington, DC, USA, 2001. IEEE Computer Society.

[111] Vanhanen J., M.V.Mantyla, and J. Ithonen, " Lightweight elicitation and analysis of software product quality goals a multiple industrial case study," Third International Workshop on Software Product Management (IWSPM), pp. 42-52, September 2009.

[112] vensson, R.B.; Gorschek, T.; Regnell, B.; Torkar, R.; Shahrokni, A.; Feldt, R.; Aurum, A., "Prioritization of quality requirements: State of practice in eleven companies," Requirements Engineering Conference (RE), 2011 19th IEEE International , vol., no., pp.69,78, Aug. 29 2011-Sept. 2 2011

[113] Wagner, S.; Deissenboeck, F., "An Integrated Approach to Quality Modelling," Software Quality, 2007. WoSQ'07: ICSE Workshops 2007. Fifth International Workshop on , vol., no., pp.1,1, 20-26 May 2007

[114] Wei B., Z. Jin, and L. Liu, "A formalism for extending the NFR Framework to support the composition of the goal trees,"in 17th Asia Pacific Software Engineering Conference, 2010, pp. 23-32

[115] Wei B.and Z. Jin, "Characterizing the implementation of software non-functional requirements from probabilistic perspective,"in COMPSAC, 2011, to appear.

[116] Wen C. Too, Sa'adah Hassan, Jamilah Din, and Abdul Azim Abd. Ghani, Towards Improving NFR Elicitation in Software Development, International Journal of Information Technology & Computer Science (IJITCS) (ISSN No : 2091-1610)Volume 7 : No : 1 : Issue on January / February, 2013

[117] Xu Lihua , Hadar Ziv, Debra Richardson, and Zhixiong Liu. Towards Modeling Non-Functional Requirements in Software Architecture. 2005.

[118] Ye Fei; Zhu Xiaodong, "An XML-Based Software Non-Functional Requirements Modeling Method," Electronic Measurement and Instruments, 2007. ICEMI '07. 8th International Conference on , vol., no., pp.2-375,2-380, Aug. 16 2007-July 18 2007

[119] Yin Bin , Jin Zhi, Chen Xiaohong An approach for selecting implementation strategies of non-functional requirements in Proceedings of the Fourth Asia-Pacific Symposium on Internetware Article No. 20 ACM New York, NY, USA ©2012.

[120] Yrjönen A, Merilinna J (2010) Tooling for the full traceability of non-functional requirements within model-driven development. In: 6th ECMFA traceability workshop (ECMFA-TW’10), Paris, 15 June 2010

[121] Yu E., "Towards modelling and reasoning support for early-phase requirements engineering," Proc. of the 3rd IEEE International Symposium on Requirements Engineering (RE' 97), pp. 226-235, 1997.

[122] Zheng X.; Xiaomei Liu; Shulin Liu, "Use Case and Non-functional Scenario Template-Based Approach to Identify Aspects," Computer Engineering and Applications (ICCEA), 2010 Second International Conference on , vol.2, no., pp.89,93, 19-21 March 2010

[123] Zhu Liming and Ian Gorton. UML Profiles for Design Decisions and Non- Functional Requirements. In 2ndIntl. Workshop on SHAringand Reusing architectural Knowledge, 2007.

[124] Zhu Ming-Xun , Xin-Xing Luo , Xiao-Hong Chen , Desheng Dash Wu, A non-functional requirements tradeoff model in Trustworthy Software, Information Sciences: an International Journal, 191, p.61-75, May, 2012

[125] Zou J. and C.J. Pavlovski, "Modeling Architectural Non Functional Requirements: From Use Case to Control Case", Proc. of IEEE International Conference on e-Business Engineering (ICEBE '06), Shanghai China, 2006, pp. 315-322.

[126] Zschaler S., "Towards a Semantic Framework for Nonfunctional Specifications of Component-Based Systems", Proc. 30th EUROMICRO Conf., Rennes, France, Aug./Sep. 2004, pages 92-99, IEEE Computer Science, Sept. 2004.

[127]Zschaler, S.: Formal specification of non-functional properties of component-based software systems: A semantic framework and some applications thereof. Software and Systems Modelling (SoSyM) 9, 161–201 (2009)