Idea: A Feasibility Study in Model Based Prediction of Impact of Changes on System Quality

Aida Omerovic[1,2], Anette Andresen[3], Håvard Grindheim[3], Per Myrseth[3],
Atle Refsdal[1], Ketil Stølen[1,2], and Jon Ølnes[3]

[1] SINTEF ICT, Norway
[2] University of Oslo, Norway
[3] DNV, Norway

Abstract. We propose a method, called PREDIQT, for model based prediction of impact of architecture design changes on system quality. PREDIQT supports simultaneous analysis of several quality attributes and their trade-offs. This paper argues for the feasibility of the PREDIQT method based on a comprehensive industrial case study targeting a system for managing validation of electronic certificates and signatures worldwide. We give an overview of the PREDIQT method, and present an evaluation of the method in terms of a feasibility study.

1 Introduction

When adapting a system to new usage patterns or technologies, it is necessary to foresee what such adaptions of architectural design imply in terms of system quality. Predictability with respect to non-functional requirements is one of the necessary conditions for the trustworthiness of a system.

We have developed the PREDIQT method with the aim to facilitate prediction of impacts of architecture design changes on system quality. The PREDIQT method produces and applies a multi-layer model structure, called prediction models, which represent system design, system quality and the interrelationship between the two. Our overall hypothesis is that the PREDIQT method can, within practical settings and with needed accuracy, be used to predict the effect of specified architecture design changes, on the quality of a system. Quality is decomposed through a set of quality attributes, relevant for the target system. PREDIQT supports simultaneous analysis of all the identified quality attributes and their trade-offs. The PREDIQT approach of merging quality concepts and design models into multiple, quality attribute oriented "Dependency Views" (DVs), and thereafter simulating impacts of architecture design changes on quality, is novel.

This paper reports on experiences from using the PREDIQT method in a major industrial case study focusing on a so-called "Validation Authority" (VA) system [10] for evaluation of electronic identifiers worldwide. We first give an overview of the PREDIQT method, and then present an evaluation of the method

F. Massacci, D. Wallach, and N. Zannone (Eds.): ESSoS 2010, LNCS 5965, pp. 231–240, 2010.

in terms of a feasibility study. The evaluation focuses on security and its trade-offs with the overall quality attributes identified and defined with respect to the VA system, namely scalability and availability. The results indicate that PREDIQT is feasible in practice, in the sense that it can be carried out on a realistic industrial case and with limited effort and resources.

The paper is organized as follows: An overview of the PREDIQT method is provided in Section 2. Section 3 presents the feasibility study. Section 4 provides an overview of related research. Concluding remarks are summarized in Section 5. See the full technical report [12] for a more detailed presentation.

2 Overview of the the PREDIQT Method

The process of the PREDIQT method consists of the three overall phases illustrated by Figure 1. Each of these phases is decomposed into sub-phases. Sections 2.1 and 2.2 present the "Target modeling" and "Verification of prediction models" phases, respectively. The "Application of prediction models" phase consists of the sub-phases presented in Section 2.3.

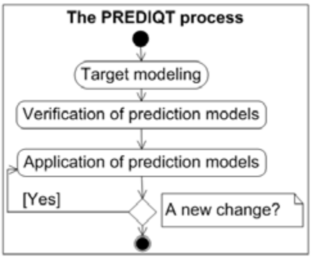
The PREDIQT process
Target modeling
Verification of prediction models
Application of prediction models
[Yes]
A new change?

Fig. 1. The overall PREDIQT process

2.1 Target Modeling

The sub-phases within the "Target modeling" phase are depicted in Figure 2. The requirement specifications and system design models are assumed to be made available to the analysis team, along with the intended usage, operational environment constraints (if available) and expected nature and rate of changes.

Characterize the target and the objectives: Based on the initial input, the stakeholders involved deduce a high level characterization of the target system, its scope and the objectives of the prediction analysis, by formulating the system boundaries, system context (including the operational profile), system life time and the extent (nature and rate) of design changes expected.

Create quality models: Quality models are created in the form of a tree, by decomposing total quality into the system specific quality attributes and their respective sub-characteristics. The quality models represent a taxonomy with interpretations and formal definitions of system quality notions.

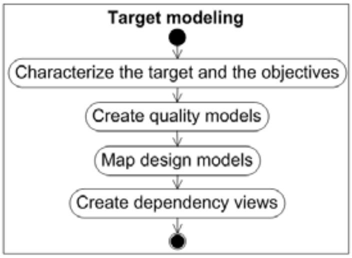

Fig. 2. Target modeling phase

Map design models: The initially obtained design models are customized so that (1) only their relevant parts are selected for use in further analysis; and (2) a mapping within and across high-level design and low-level design models (if available), is made. The mapped models result in a class diagram which includes the relevant elements and their relations only.

Create dependency views: In order to ensure traceability to (and between) the underlying quality models and the mapped design model, a conceptual model (a tree-formed class diagram) where classes represent elements from the underlying design and quality models, relations show the ownership, and the class attributes indicate the dependencies, interactions and the properties, is created. The conceptual model is transformed into a generic DV – a directed tree representing relationships among quality aspects and design of the system. For each quality attribute defined in the quality model, a quality attribute specific DV is created, by a new instantiation of the generic DV. Each set of nodes having a common parent is supplemented with an additional node called “Other”, for completeness purpose. In addition, a total quality DV is deduced from the quality models. The DV parameters are evaluated by providing the estimates on the arcs and the leaf nodes, and propagating them according to a pre-defined model.

2.2 Verification of Prediction Models

The set of preliminary prediction models developed during the “Target modeling” phase, consists of design models, quality models and the DVs. The “Verification of prediction models” phase (Figure 3) aims to validate the prediction models (with respect to the structure and the individual parameters), before they are applied.

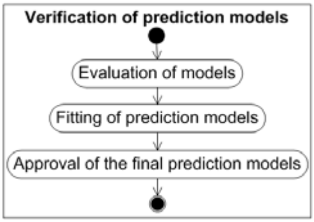
Verification of prediction models
Evaluation of models
Fitting of prediction models
Approval of the final prediction models

Fig. 3. Verification of models – phase

Evaluation of models: A measurement plan with the necessary statistical power is developed, describing what should be evaluated, when and how.

Fitting of prediction models: Model fitting is conducted in order to adjust the DV parameters to the evaluation results.

Approval of the final prediction models: The objective of this sub-phase is to evaluate the prediction models as a whole and validate that they are complete, correct and mutually consistent after the fitting. If the deviation is above the acceptable threshold after the fitting, the target modelling is re-initiated for structural model changes.

2.3 Application of Prediction Models

During the “Application of prediction models” phase (depicted by Figure 4), a specified change is applied and simulated on the approved prediction models.

Specify a change: The change specification should clearly state all deployment relevant facts, necessary for applying the change.

Apply the change on prediction models: This phase involves applying the specified architecture design change on the prediction models. See [12] for further details.

Quality prediction: The propagation of the change throughout the rest of each one of the modified DVs, as well as the total quality DV, is performed based on the general DV model. See [12] for further details.

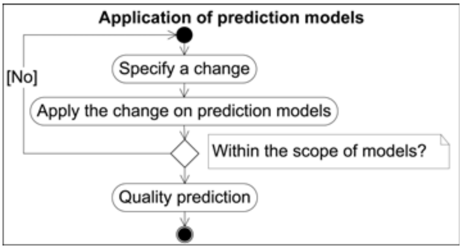
Application of prediction models
Specify a change
[No]
Apply the change on prediction models
Within the scope of models?
Quality prediction

Fig. 4. Application of models – phase

3 Application of PREDIQT in the Industrial Case

The PREDIQT method was tried out in a major industrial case study focusing on a Validation Authority (VA) [9] system.

3.1 Target Modeling

Based on the initially obtained documentation (requirement specifications, operational environment specifications etc.) and several interactions with the domain experts, UML based design models of the VA were developed.

Characterize the target and the objectives: An overview of the functional properties of the VA, the workflows it is involved in and the potential changes, was provided, in order to determine the needed level of detail and the scope of our prediction models. Among the assumed changes were increased number of requests, more simultaneous users and additional workflows that the VA will be a part of. The target stakeholder group is the system owner.

Create quality models: An extract of the quality model of the VA is shown by Figure 5. Total quality of the VA system is first decomposed into the four quality attributes: availability, scalability, security and "Other Attr." (this node covers the possibly unspecified attributes, for model completeness purpose). The X, Y, Z and V represent system quality attribute ratings, while u,v g and j represent the normalized weights expressing each attribute's contribution to the total quality of VA. The attributes and their ratings are defined with respect to the VA system, in such a manner that they are orthogonal and together represent the total quality of the VA system. Thereafter, the sub-characteristics of each quality attribute are defined with respect to the VA system, so that they are orthogonal to each other and represent a complete decomposition of the VA attribute in question. Both the attribute ratings and the sub-characteristic measures are defined so that their value lies between 0 (minimum) and 1 (maximum fulfillment).

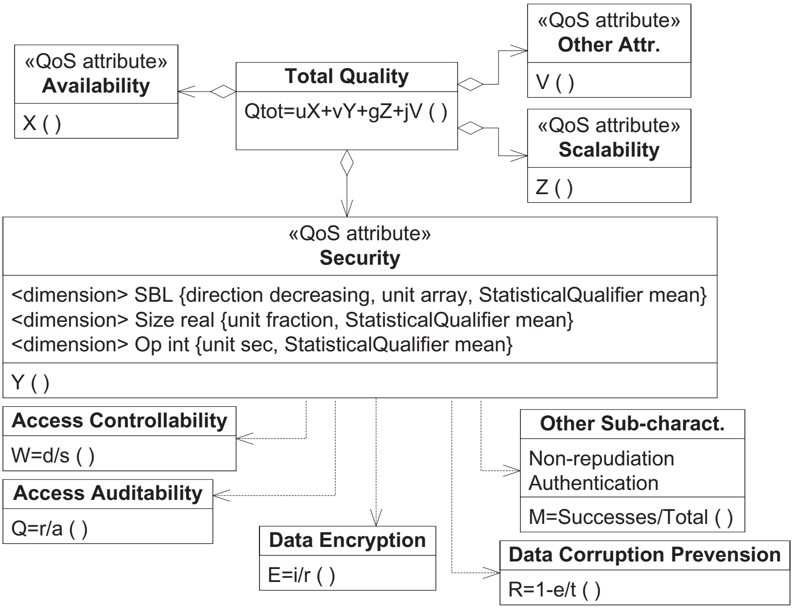

Fig. 5. An extract of the quality model

Consider the attribute "Security" in Figure 5. Its weight is v (the coefficient of Y in the "Total Quality" class). The definition of the "Security" attribute was based on [2] and its rating was based on [11]. The dimensions-notation shown in the form of attributes of the Security class in Figure 5, originates from [3]. The dimensions represent the variables constituting the rating of the Security attribute for the VA system. Given the attribute and rating definition, the appropriate (for the attribute and the VA system) sub-characteristics were retrieved from [1], where they are fully defined. The Q,W,E,R,M represent the formal measure definitions of each sub-characteristic. Security depends on the five sub-characteristics displayed: access auditability, access controllability, data encryption, data corruption prevention and "other sub-characteristics". Access auditability, for example, expresses how complete the implementation of access login is, considering the auditability requirements. Its measure is: Access auditability $= \frac{r}{a}$, where r = Nr. of information recording access log confirmed in review; a = Nr. of information requiring access log [1].

Map design models: The originally developed design models of the VA covered both high-level and low level design aspects through use cases, class diagrams, composite structure diagrams, activity diagrams and sequence diagrams. A selection of the relevant models was made, and a mapping between them was undertaken. The model mapping resulted in a class diagram containing the selected elements (lifelines, classes, interfaces etc.) as classes, ownership as relations and interactions/dependencies/properties as class attributes. Due to only ownership representing a relation, this resulted in a tree-formed class diagram.

Create dependency views: A conceptual model (a tree-formed class diagram) with classes representing the selected elements, relations denoting ownership, and selected dependencies, interactions quality properties, association relationships or their equivalents represented as the class attributes, is deduced from (1) the quality models; and (2) the above mentioned class diagram. See Figure 6 for an extract. Further details on the construction of the conceptual model are provided in [12].

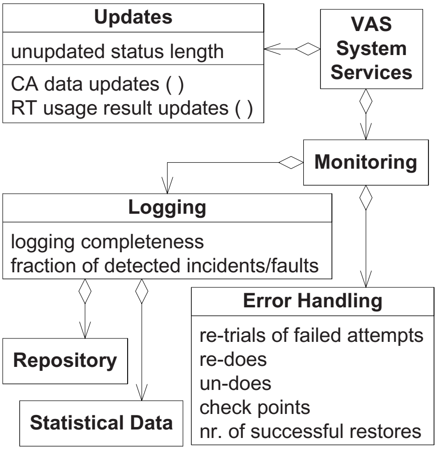

Fig. 6. An extract of the conceptual model

A generic DV – a directed tree representing relationships among quality aspects and design of the system was obtained by instantiating the conceptual model. Quality attribute specific DVs were derived in the form of directed trees with the structure from the generic DV. Each set of nodes having a common parent was supplemented with an additional node called "Other", for completeness purpose. In addition, a total quality DV was instantiated from the top two levels of the quality models.

The DVs were inserted into the tool we have built on top of MS Excel for enabling automatic simulations within and across the DVs. A QCF denotes the "degree of Quality attribute or Characteristic Fulfillment" and is associated with each node of a DV. The QCF values of attribute specific DVs were estimated by assigning a value of the quality attribute (which the DV under consideration is dedicated to) to each leaf node of the quality attribute specific DVs. The QCF value quantification involved revisiting quality models, and providing a quality attribute rating value to each node. The QCF value expresses to what degree the quality attribute (given its system specific formal definition from the quality models) represented by the DV is fulfilled within the scope of the node in question. Due to the rating definitions, the values of QCFs are constrained between 0 (minimum) and 1 (maximum). An EI denotes the "Estimated Impact" and is associated with each arc of the DVs. EI expresses the degree of impact of a child node (which the arc is directed to) on the parent node, or to what degree the parent node depends on a child node. An EI value is assigned with respect to the sub-characteristics of the quality attribute under analysis (defined in the quality models) and their respective impact on the relation in question. The EI on each arc was assigned by evaluating the impact of the child node on its parent node, with respect to the sub-characteristics (defined in the quality models) of the attribute under consideration. Once a sum of the contributions of the sub-characteristics was obtained on each arc pointing to children nodes with a common parent, the EI values were normalized so that they sum up to 1 (due to model completeness).

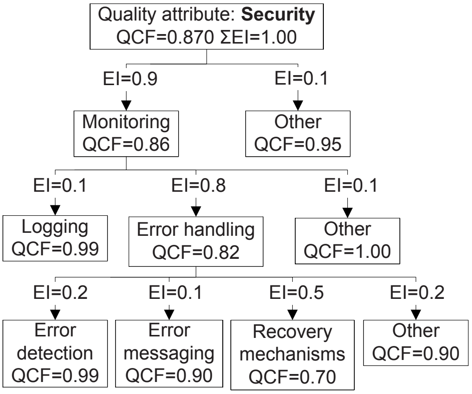

Fig. 7. A part of the Security DV

Figure 7 shows an extract of the Security attribute specific DV of VA (the values assigned are fictitious, for confidentiality reasons). In the case of the "Error detection" node of Figure 7, the QCF value expresses the effectiveness of error detection with respect to security. The QCFs as well as the EIs of this particular DV are estimated with reference to the definition of Security attribute and its sub-characteristics, respectively. The definitions are provided by the quality models exemplified in Figure 5. The total quality DV is assigned weights on the arcs, which, based on the attribute definitions in the quality models, express the impact of each attribute (in terms of the chosen stakeholder's gain or business criticality), on the total system quality. The weights are system general objectives. The weights are normalized and sum up to 1, since also this DV is complete. The leaf node QCFs of the total quality DV correspond to the root node QCFs of the respective quality attribute DVs. Once estimates of leaf nodes' QCFs and all EIs are provided, the QCF values of all the non-leaf nodes are automatically inferred by the tool.

3.2 Verification of Prediction Models

Verification of the prediction models for the VA relied on measurements and expert judgments. The details are provided in Section 3.2 of [12].

3.3 Application of Prediction Models

The prediction models were applied for simulation of impacts of 14 specified, independent architecture design changes on the VA quality. Each specified architecture design change was first applied on the affected design models, followed by the conceptual model and finally the DVs. Application of a change on each quality attribute specific DV involved:

1. The DV structure was modified in order to maintain consistency with the modified conceptual model (which maps the design and the quality models).
2. For those leaf nodes that were directly traceable to the affected attributes (which represent properties, interactions and dependencies in the design models) of the conceptual model illustrated by Figure 6, the leaf node QCFs were modified by the analyst (based on the quality attribute rating definition) if appropriate for the DV in question (recall the quality model).

3. The affected arcs were identified, based on the affected attributes of the conceptual model (illustrated by Figure 6). The EI values on the affected arcs were changed by the analyst, by re-evaluating the impact of the sub-characteristics of the attribute that the DV is dedicated to and normalizing them on the arcs pointing to the nodes having a common parent. Which DVs were affected and the extent of modification of the identified EIs on them, was determined by the definitions from the quality models.
4. The modifications and their rationale were documented.

The propagation of the changes throughout and across the DVs is performed based on the general DV model, according to which the QCF value of each parent node is recursively calculated by first multiplying the QCF and EI value for each closest child and then summing these products. The DVs and the model were embedded into the above mentioned tool, which allowed simulating the change propagation in an “what-if” manner. See [12] for further details.

4 Related Work

[13] presents risk identification techniques like principal component analysis, discriminant analysis, tree-based reliability models and optimal set reduction. Common for all these techniques is that they analyze and, to some degree, predict defects, based on low-level data. [3] provides a UML notation for QoS modelling, which has been applied in our quality models. PREDIQT is also compatible with the established software quality standard [1]. The goal/question/metric paradigm [5] [4] is also compatible with PREDIQT and can be used for development of quality models and design of a measurement plan [6]. [8] and [14] introduce approaches to pattern based quantitative security assessment. Both are solely security oriented approaches for security assessment (and not prediction) with limited traceability to the system design and quality notions. [7] argues that traditional statistical approaches are inadequate and recommends holistic models for software defect prediction, using Bayesian Networks. However, a drawback that statistical and BBN-based models suffer, is poor scalability.

5 Conclusions and Future Work

This paper has presented PREDIQT – a method for model based prediction of impacts of architecture design changes on system quality. We have also reported on results from applying PREDIQT in a major industrial case study. The case study focused on prediction of security and its trade-offs with the overall quality attributes of the target system. We basically did two evaluations, the feasibility study described above and a thought experiment.

In relation to the feasibility study, a lightweight post-mortem analysis was conducted. The domain experts, who participated in the analysis, expressed that the development of DVs was relatively simple, thanks to the comprehensible DV models as well as the confined underlying quality and design models. One of the main points of the feedback was that the reasoning around DVs, particularly their parametrization, facilitated understanding the system design and reasoning about alternatives for potential improvements, as well as existing and potential weaknesses of architectural design, with respect to the quality attributes. We managed to conduct all the steps of the PREDIQT method, with limited resources and within the planned time period (six half-a-day workshops with upto one man-labour week before each). The changes specified were deduced with the objective of addressing the most relevant parts of the prediction models, being diverse and realistic. The fact that all the 14 specified changes were within the scope of the prediction models and could be simulated within the PREDIQT method, indicates feasibility of developing the prediction models with intended scope and quality. Overall, applying PREDIQT was feasible within the practical settings of this case. The models were relatively straight forward to develop, and judged to be fairly easy to use.

The predictions obtained were evaluated by means of a thought experiment on a domain expert panel with thorough knowledge of the VA system. The process and the results from simulation of the 14 specified changes in relation to the feasibility study were obtained and documented by the analysis leader, stored by an additional analysis participant, and kept unrevealed. Independently, the domain experts were asked to estimate the impact of each change on the respective quality attributes defined. The results obtained show quite a low magnitude of deviation between the PREDIQT based simulations and the values obtained from the thought experiment. We do not have hard evidence that the predictions were correct, but given the research method and the values obtained, the results of the thought experiment are promising. Further details on the PREDIQT method, the feasibility study and the thought experiment are presented in [12].

We expect PREDIQT to be applicable in several domains of distributed systems with high quality and adaptability demands. Handling of inaccuracies in the prediction models, improving traceability of the models and design of an experience factory, are among the partially initiated future developments.

Acknowledgments. This work has been conducted within the DIGIT (180052/S10) project, funded by the Research Council of Norway.

References

1. International Organisation for Standardisation: ISO/IEC 9126 - Software engineering – Product quality (2004)
2. ISO/IEC 12207 Systems and Software Engineering – Software Life Cycle Processes (2008)
3. Object Management Group: UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms, v. 1.1 (2008)
4. Basili, V., Caldiera, G., Rombach, H.: The Goal Question Metric Approach. In: Encyclopedia of Software Engineering. Wiley, Chichester (1994)
5. Basili, V.R.: Software Modeling and Measurement: the Goal/Question/Metric Paradigm. Technical Report TR-92-96, University of Maryland (1992)

6. Ebert, C., Dumke, R., Bundschuh, M., Schmietendorf, A., Dumke, R.: Best Practices in Software Measurement. Springer, Heidelberg (2004)
7. Fenton, N., Neil, M.: A Critique of Software Defect Prediction Models. IEEE Transactions on Software Engineering 25, 675–689 (1999)
8. Heyman, T., Scandariato, R., Huygens, C., Joosen, W.: Using Security Patterns to Combine Security Metrics. In: ARES 2008: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security, Washington, DC, USA, pp. 1156–1163. IEEE Computer Society, Los Alamitos (2008)
9. Ølnes, J.: PKI Interoperability by an Independent, Trusted Validation Authority. In: 5th Annual PKI R&D Workshop, NIST Gaithersburg MD, USA (2006)
10. Ølnes, J., Andresen, A., Buene, L., Cerrato, O., Grindheim, H.: Making Digital Signatures Work across National Borders. In: ISSE 2007 Conference: Securing Electronic Business Processes, pp. 287–296. Warszawa, Vieweg Verlag (2007)
11. Omerovic, A.: Design Guidelines for a Monitoring Environment Concerning Distributed Real-Time Systems. Tapir Academic Press, Trondheim (2004)
12. Omerovic, A., Andresen, A., Grindheim, H., Myrseth, P., Refsdal, A., Stølen, K., Ølnes, J.: A Feasibility Study in Model Based Prediction of Impact of Changes on System Quality. Technical report, SINTEF A13339 (2009)
13. Tian, J.: Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, 1st edn. Wiley-IEEE Computer Society Press, Chichester (2005)
14. Yautsiukhin, A., Scandariato, R., Heyman, T., Massacci, F., Joosen, W.: Towards a Quantitative Assessment of Security in Software Architectures. In: 13th Nordic Workshop on Secure IT Systems, Copenhagen, Denmark (2008)