Systematic Selection of Quality Attribute Techniques

Yin Kia Chiam
NICTA, Sydney, Australia
School of CSE, University of New South Wales, Sydney, Australia
FCSIT, University of Malaya, Kuala Lumpur, Malaysia
yinkia.chiam@nicta.com.au

Liming Zhu
NICTA, Sydney, Australia
School of CSE, University of New South Wales, Sydney, Australia
liming.zhu@nicta.com.au

Mark Staples
NICTA, Sydney, Australia
School of CSE, University of New South Wales, Sydney, Australia
mark.staples@nicta.com.au

ABSTRACT
Various techniques are used to investigate, evaluate, and control product quality risks throughout software development process. These “Quality Attribute Techniques” are used during all stages of the software development life cycle to ensure that acceptable levels of product qualities such as safety and performance are in place. In this paper, we propose a method to select from among the alternatives of these techniques. This method is based on Risk Management theory and the Analytic Hierarchy Process (AHP) approach. We apply our method to an example of real-world safety system presented in the literature. We identify advantages and limitations of the method, and discuss future research.

Categories and Subject Descriptors
D.2.9 [Software Engineering]: Management—life cycle, software quality assurance; K.6.3 [Management of Computing and Information Systems]: Software Management—software process

General Terms
Management, Theory

Keywords
Quality Attribute Techniques, Product Quality, Software Process Improvement, Selection Method

1. INTRODUCTION
The IEEE Standard Glossary of Software Engineering Terminology[8, p.60] defines software product quality as “the degree to which a system, component, or process meets specified requirements, customer or user needs or expectations”. Product quality risks need to be managed throughout the software development process to achieve product quality requirements. We define product quality risk as the potential quality problems that may cause a system to fail to meet its specified level of quality requirements.

Various techniques exist to investigate, evaluate and control product quality risks (PQR). In this paper, we call these “Quality Attribute Techniques” (QAT). QATs are used in software development to achieve product qualities such as safety and security. Examples of QATs for safety include hazard analysis techniques such as Failure Mode and Effect Analysis (FMEA) and Fault Tree Analysis (FTA). Process improvements to identify, analyse, and control PQRs could be undertaken by integrating corresponding QATs into the appropriate development phases or activities. However, the research literature has not normally regarded PQRs as an important characteristic for software process tailoring. Although QATs are used in practice, they are not currently represented in detail or incorporated well in software development process models [15, 16].

QATs can range from relatively simple and cheap walk-throughs and checklists, to more complex and expensive approaches requiring intensive expertise such as Fault Tree Analysis and formal methods. Development teams need to select appropriate QATs and incorporate them into development processes created or tailored for new projects. Most prior research has focused on selecting QATs for individual quality attribute (e.g. safety, performance) or specific life cycle phases (e.g. requirement, architecture). Some selection methods are only suitable for verification and validation QATs and are not suitable for other types of QATs.

This paper proposes and evaluates a systematic selection method for comparing and choosing QATs for any quality attribute, across the life cycle. The method is intended to help development teams select QATs and integrate them into the life cycle. We use risk management as a general theory to encompass a variety of product qualities. A multi-criteria decision making method, AHP, is applied to analyse the cost/benefit of QAT alternatives. The focus of this paper is the research question: How can appropriate QATs be selected for inclusion in a tailored software process that targets a specific product quality attribute?

The structure of this paper is as follows. Section 2 provides background on risk management, QAT and AHP. Section 3 describes the proposed selection method. Section 4 discusses advantages and limitations of the selection method arising from the evaluation on an exemplar. Section 5 concludes and discusses future research.
2. BACKGROUND

2.1 Risk Management for Product Quality Risks
According to AS/NZS 4360[3, p.4], risk management is “the culture, processes and structures that are directed towards realizing potential opportunities whilst managing adverse effects”. Risk management is usually applied as a project management discipline, but in this paper we apply it to manage product qualities. Previously proposed process tailoring methods do not use a risk-based approach to manage a variety of product qualities.

We define PQRs as the potential quality problems that may cause a system to fail to meet the specified level of quality requirements. We categorise the QATs in order to help development teams to select QATs based on the risk management theory. After defining quality objectives, PQRs can be identified. The impact of each PQR can be analysed and ranked according to its probability of occurrence and severity of damage. The amount of effort to eliminate or prevent specific PQRs can be determined by the level of risk.

2.2 QAT Overview
QATs are used to identify, analyze and control PQRs in the development. For instance, safety critical systems are concerned with hazards to life, property or the environment, and performance-critical systems emphasize response time or throughput. QATs are usually technical engineering techniques [15] that target specific product quality attributes. Examples of QATs for safety include hazard analysis techniques and performance-critical systems emphasize response time or throughput. QATs are usually technical engineering techniques such as Failure Mode and Effect Analysis (FMEA) and Fault Tree Analysis (FTA). However, despite their importance in practice, QATs are not usually represented in detail in software development process models.

2.3 Analytic Hierarchy Process (AHP)
Numerous multicriteria decision making (MCDM) methods have been developed to help solving decision problems by evaluating a set of alternatives against pre-specified criteria. In this research, we apply AHP to evaluate the alternatives of QATs according to cost/benefit criteria. AHP is widely used in industry and academia, for example in software requirements prioritisation [9] and software architecture evaluation [2, 14].

Unlike other MCDM methods, the decision making process in AHP is based on relative assessment. In AHP, all alternatives can be evaluated using pair-wise comparisons. As a result, the evaluation is less sensitive to judgment errors when compared to other MCDM methods using absolute assignments [2, p.246]. However, the main limitation of AHP is scalability when there are many alternatives. To mitigate this problem, we combine basic filtering with AHP, based on risk management. AHP is used to evaluate and select QATs from shortlisted alternatives.

3. QAT SELECTION METHOD
We propose a method to help development teams select QATs for any quality domain. This method compares the QAT alternatives from three perspectives: risk management, process integration, and cost/benefit. For the cost/benefit perspective, AHP is applied to evaluate and compare these alternatives. We have previously developed a QAT Framework (QATF) [6] to capture information about QATs. We compiled an example list of safety QATs by reviewing the software literature in safety area.

3.1 Risk Management Perspective
We use risk management to understand how QATs function to affect product quality by identifying, analyzing, and controlling PQRs. We categorise the QATs according to the method by which they address PQRs. Following the AS/NZS 4360, we have grouped the QATs into two main categories: quality risk assessment and quality risk control. Quality risk assessment is the process of identifying, analysing and evaluating the PQRs. Quality risk control is the process of treating PQRs and confirming the fulfillment of product quality requirements. We describe these categories below using safety QATs as illustrative examples.

3.1.1 Quality Assessment Techniques.
1. PQR Identification - QATs which produce lists of the product-specific quality risks may compromise a product’s satisfactory outcome.
2. PQR Analysis - QATs which produce assessments of the probability and magnitude of losses associated with each of the identified PQRs, and assessments of sources, causes and consequences involved in PQRs.
3. PQR Evaluation - QATs which produce a prioritised ordering of the PQRs identified and analysed.

3.1.2 Quality Control Techniques.
1. PQR Treatment - QATs which resolve, reduce or eliminate risk items and take corrective action when appropriate.
2. PQR Monitoring - QATs to determine and monitor the quality specification of either a phase or the consistency of a complete system with the quality requirements.

This categorisation of QATs is intended to help development teams select QATs based on how these QATs manage PQRs. Additionally, development teams can compare the QAT alternatives in the same category by referring to the impact of the QATs on quality risk management.

No single QAT can handle all aspects of system quality, especially for high assurance systems. If development teams apply only one or two QATs without understanding the impact of the QATs in quality risk management, the intended product quality may not be achieved. A combination of QATs should be selected for effective PQR assessment and control across the life cycle.

3.2 Process Integration Perspective
AS/NZS ISO/IEC 12207 has been used as a basis for defining a generic software development process. Product quality risk management activities take place in every phase of the software life cycle. First, the PQR management activities are mapped to software development phase. Figure 1 shows a mapping between generic PQR management activities and software development processes. The mapping is
based on the purpose and outcome of each software development phase defined in ISO/IEC 12207 [4].

Figure 1: Mapping of Product Quality Risk Management Activities to Software Development Process

- **Requirements Elicitation**: to gather, process, and track customer needs and requirements throughout the life cycle so as to establish a requirements baseline.

- **Software Requirements Analysis**: to establish the requirements of the software elements of the system.

- **Software Design**: to provide a design for the software that implements and can be verified against the requirements.

- **Software Construction (Code and Unit Test)**: to produce executable software units that properly reflect the software design.

- **Software Integration**: to combine units into integrated software items, consistent with the software design, that demonstrate that the functional and non-functional software requirements are satisfied.

- **Software Testing**: to confirm that the integrated software product meets its defined requirements.

After matching risk management activities with software processes, development teams can select QATs which is intended to help development teams to select QATs for risk management activities and software phases. An earlier initial literature review [6] identified QATs in safety area. The review analysed the process characteristics for QATs such as inputs (the prerequisite work products), outputs (the work products created or modified), and the development process phase(s). A relevant QAT which can be applied to intended process phases or activities can be selected by referring to these process characteristics. Some QATs can be used in multiple development phases.

### 3.3 Cost/Benefit Perspective

We use AHP to evaluate the ranking of each QAT alternative under each selection criterion. The selection criteria was identified in previous work [6]. An initial review of literature relevant to software safety (e.g. [10, 5, 17]), software architecture design (e.g. [14]) and software testing technique (e.g. [13, 7]) identified high and low level criteria which influence cost/benefit of applying a QAT. The criteria are as follows:

- Cost of Application: Time required, Cost of Tool, Documentation, Test Data Cost
- Expertise: Knowledge, Experience, Training
- Number of Performer: Individual or Team, Team Size
- Impact: Failures Detected, Qualitative or Quantitative Results, Execution Time, Modifiability, Scalability, Portability, Ease of Installation, Repeatability, Test Case Completeness, Test Case Precision

AHP involves five main steps for evaluation of QAT alternatives. Four PQR assessment techniques: HAZOP, FMEA, FTA and ETA will be used as alternatives for selection in this example:

**Step 1**: Set up the $n$ selection criteria in the rows and columns of an $n \times n$ matrix. For example, if we want to select PQR assessment techniques based on four selection criteria: Cost of Application, Experience, Failures Detected and Number of Performers, we will have a 4 x 4 matrix.

**Step 2**: Weighting selection criteria using pairwise comparisons. The relative importance of one criterion over another can be expressed by comparing every pair of selection criteria using the weighting scale (1=Equal importance; 3=Moderate importance; 5=Strong importance; 7=Very strong importance; 2,4,6,8=Intermediate values between two adjacent judgement). Development teams can use their judgement to weight the relative importance of each element based on the QAT and project characteristics.

**Step 3**: Determine relative ranking of each QAT alternative over another under each criterion. For each criterion, the values of the weighting scale in the Step 2 will be used to weight the preference of each alternative over another in pairs. The information captured by QATF for each alternative is intended to help development teams to make judgement by referring to the recommendations from literature, practitioners and experts.
Step 4: Use averaging over normalized columns to estimate the eigenvalues of the pairwise matrix. Averaging over normalized columns is a method proposed by Thomas Saaty [12]. Here, we use a web-based analysis tool provided by AHPproject [1] to compute this.

Step 5: Perform sensitivity analysis. Sensitivity analysis provide information on how the alternatives ranking behaves in response to changes in priorities. This analysis can be illustrated using diagrams. We use the AHPproject analysis tool to produce diagrams for sensitivity analysis.

4. EVALUATION ON AN EXEMPLAR

We applied our method to select PQR Assessment Techniques for safety analysis using the Center-TRACON Automation System (CTAS) [11] as an exemplar project, and to compare QATs. In the project, the safety QATs had not originally been incorporated into software development process model. We used our approach to select safety QATs suitable for complex systems and integrate them into relevant software phases.

We compared our selected QATs with the actual QATs used in this project. The same QATs selected by us and used by the team are GHA and SMHA. We selected FMECA and Procedure or Task analysis, which are recommended by literature, to analyse hazards and human errors for this complex system. The team used QATs (SDA and SpecTRM-RL) and tools developed by them to support these analysis. They also used FTA to produce parts of fault tree. Instead of choosing FTA, we had selected CCA for more detailed analysis to produce cause and consequence diagrams. The project team claimed that their new methods and tools overcome the limitation of existing QATs. It is hard to consider explicitly since the methods and tools are not commonly used.

The main limitation of our proposed method is it requires judgement from the development team to make the final selection decision. The QAT information is intended to help the development teams to make judgement by understanding the QATs, comparing the QATs and also analyse the relationships between QATs and other process elements.

5. CONCLUSIONS AND FUTURE WORK

This paper has proposed a QAT selection method to help development teams to compare, evaluate and choose appropriate QATs based on three perspectives: risk management, process integration and cost/benefit. The categorisation of QATs and process information captured by QATF are intended to help development teams to select QATs based on how these QATs manage the PQRs. AHP is applied to evaluate and compare these alternatives based on cost/benefit criteria. Our initial evaluation applied the proposed method to an exemplar safety-critical development project reported in the literature. The evaluation shows that the proposed method is feasible to apply, and that the results of the method are broadly comparable with the QATs selected by the expert process engineers. Our method also helps to determine the sequence of using the QATs by considering the relationships between QATs and other process elements.

Our future work will develop an integration framework to incorporate the selected QATs into software development process models. Further theoretical and empirical evaluation is required to more fully evaluate the proposed method.

6. ACKNOWLEDGMENTS

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

7. REFERENCES