A Framework for Software Requirements Engineering

KHAIRUDDIN HASHIM  NURUL NASLIA KHAIRUDDIN
Software Engineering Department  Faculty of Computer Science and Information Technology
Tenaga Nasional University  University of Malaya
43009 Kajang  50603 Kuala Lumpur
MALAYSIA  MALAYSIA
khairuddin@uniten.edu.my  nurul_naslia@perdana.um.edu.my

Abstract: - Despite research being done on improving the requirements engineering process, existing requirements engineering tools and techniques are still inadequate. Different tools and techniques represent different foci on aspect of features. This paper presents a requirements engineering framework that provides emphasis on four aspects of requirements engineering – organizational factors, requirements attributes, processes and standards; and tool support.

Key-Words: - Requirements Engineering, Organizational Factors, Framework, Processes, Standards, Tool Support

1 Introduction
Many requirements engineering tools and techniques have been developed. Each approach has different features. The requirements engineering process needs cooperation from the stakeholders, especially during the elicitation of the requirements and also in the process of requirements analysis [1].

The process of requirements specification should ensure that requirements are complete, error-free, consistent and unambiguous. Good documentation of the requirements is needed to ensure ease of cross-referencing. It would also be better if ample resources could be made available in the requirements engineering process. Requirements validation is critical in ensuring that changes do not take place due to erroneous specification of requirements.

This paper proposes a requirements engineering framework with emphasis on four aspects of requirements engineering – organizational factors, requirements attributes, processes and standards; and tool support.

2 Background
In the early 1980s, researchers started to realize the importance of requirements engineering. They started to make new and clear definitions of requirements engineering terminologies. In 1990s, many researchers focus on improving the requirements engineering process. Many books were also produced by these researchers to help requirements engineering practitioners in understanding more about concepts and approaches in requirements engineering. A lot of techniques and methodologies were proposed intending to improve the quality of software requirements.

[1] described four methods for requirements engineering. The four methods are data-flow modelling, semantic data models, system modelling, and formal methods. According to them, there is no single ideal requirements approach but it is good to use more than one. [2] identified four key requirements engineering techniques. The four techniques are installing an effective core team for each product release, focusing the product life cycle on upstream gate reviews, evaluating requirements from various perspectives; and assuring dependable portfolio visibility and release implementation. According to the report, the techniques can help to significantly reduce delays.

There are also techniques proposed for particular phases of requirements engineering. A research [3] done to improve requirements tracing, framed the tracing as an information retrieval problem. A few algorithms were compared, selected and modified to implement tracing.

Research work in [4] modelled functional requirements based on use case diagrams. A modelling language, Alloy, allows for software model specification and analysis. A tool suite [5] was implemented to support the Aspect-Oriented Requirements Engineering. The tools support
identification of aspectual requirements and their influences on other requirements, conflict detection and resolution between aspectual requirements; and requirements representation and document structuring. The tool suite employs Natural Language Processing (NLP) in order to support the requirements engineer in identification of both crosscutting and non-crosscutting concerns.

Effort in integrating requirements engineering techniques is demonstrated in [6]. The research work presented 3 integration approaches highlighting advantages and disadvantages. Other research done include surveys [7], evaluations [8], and comparisons [9] of requirements engineering techniques.

3 The Framework
Software requirements engineering is an important component in the development of quality software products. The scope of requirements engineering covers:

- Elicitation – collecting requirements
- Analysis – analysing and modelling
- Specification – documenting the requirements
- Validation – checking that specification matches requirements

The challenge of making available adequate methods and tools prompted us to analyze and propose a requirement engineering framework. The framework helps emphasize critical components in the requirement engineering phase. The framework is divided into 4 components:

- Organizational Factors
- Requirements Attributes
- Processes and Standards
- Tool Support

Organizational Factors
This component covers, amongst others, managerial factors, culture, people, domain and training. Culture plays an important role in putting the practice of good management and interactions. A good culture is able to dominate positive development in an organization. Management and staff usually will be able to work as an effective team if the culture is good. A good culture of quality and continuous improvement ensures that processes are reviewed from time to time to improve on effectiveness and efficiency. This can influence the development of policies that represent decisions that allow for consistency in management. This includes training policies for staffs on technical and managerial aspects. Staffs of an organization with the right attitude represent human resource which can be nurtured to adhere to policies and procedures, maximizing on productivity. A culture of excellence allows for attainment of maturity levels. In essence, good coverage of organizational factors will ensure that the underlying foundation becomes basis for stability.

Requirements Attributes
Much of the success in specifying the user requirements depends on the attributes of the requirements. The requirements must be correct with no errors, complete covering various aspects of the requirements, consistent in terms of definition and use of requirements terms and unambiguous, which would otherwise cause multiple interpretations.

Traceability is an important attribute of requirements which allows for forward and backward tracking from source to codes and the reverse. Other attributes include feasibility, relevancy and testability. Effort also has to be given in the assessment of non-functional requirements, especially on reliability and operational performance. Coverage of important requirements attributes and the mechanism to assess them will help improve further the domain of requirements engineering.

Processes and Standards
A well defined process helps define what it takes to get a job done. Processes help break down bigger tasks into manageable ones. They help maintain work quality through quality assurance. They use resources, subject to a set of constraints. A good requirements management system is made up of well documented processes that help make process steps effective and consistent. This allows for good design and documentation including check-pointing and check-listing steps.

Standards are established and prescribed criteria and methods. Standards allows for consistency in delivery of design, document format and standard operating procedures. For example, documentation
standard specify form and content. A standard content structure will ensure that requirements are complete without missing components, for example, content structure can include trigger functions, exceptional cases and constraints which are sometimes left-out unintentionally. Advance processes include measurements on selected metrics.

Software quality assurance (SQA) represents parts of a standard approach in developing software. The rationale is that the consistent use of quality procedures will give rise to the production of quality products consistently. The SQA Plan for instance, documents the evaluations to be performed including audits, reviews and measurements. These usually include procedures for error reporting and tracking; and artifacts to be produced – what and when.

Implementing software quality assurance ensures information about software defects is collected, categorised and traced to causes. This allows for reduction of defects for current projects and ability to predict potential defects in future projects. Continuous process improvement is an essential step in ensuring effective process execution. In a nutshell, availability of processes and standards ensure consistent execution of processes which in turn delivers the required quality levels.

Tool Support
Tool support helps put conformance to a particular method through structuring, sequencing and documentation processes. Use of tools help provide support functions such as traceability and error-checking features usually found in CASE Tools. The existence of integrated CASE tools helps make required changes once through the use of a single repository.

Change management is done effectively through use of tools maintaining traceability of information including creation of knowledge structures for keeping records of rationale on decisions made. The existence of a workflow system or tool helps provide sequential or concurrent control in terms of file checking or validation procedure.

The existence of collaboration tools help facilitate multi-sites software engineering which is becoming quite common. Productivity-enhanced approaches such as software reuse is very much facilitated through the use of effective tools to help identify, catalogue and retrieve reusable tools. In essence, availability of tool support will increase productivity.

4 Conclusion
This paper documents a framework for supporting the 4 requirements engineering processes – elicitation, analysis, specification and validation. The four components of the framework which are organizational factors, requirements attributes, processes and standards; and tool support highlight areas where method and tool support can be extended further to improve on the requirements engineering environment. The framework will be extended to include missing method and tool attributes.

References: