IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Meta-Requirement Mapping Model

To cite this article: Muhamad Idaham Umar Ong and Mohamad Ariff Ameeden 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **769** 012051

View the article online for updates and enhancements.

Meta-Requirement Mapping Model

Muhamad Idaham Umar Ong[1,*]**, Mohamad Ariff Ameeden**[1]

[1]Faculty of Computing, College of Computing and Applied Sciences, University Malaysia Pahang, Gambang, 26300, Malaysia

* Correspondence: idaham@ump.edu.my

Abstract. The challenges faced by software vendor in competing for tenders in todays are massive and are desperately in need of intervention by the kind people of the research community. With very small resources available at their disposal, it is a daunting task them in the pursue to gain new software development projects each time new tenders are coming out. The objective of this publication is to propose an initial model that utilizes meta-requirement as a medium to validate the User Requirement Specification document that is included in the RFT. Looking forward, an algorithm will be developed for realizing the associations that was define in the propose model.

1. Introduction

For software vendors, software project doesn't just come falling from the sky. For high profile software project to exist, an intensive and complex task related to procurement is being executed by software purchaser. At the end of the journey, a Request for Tender (RFT) is produced. Now onwards, comes the challenging task faced by the software vendor to produce software project proposal that will be able to convince the software purchaser to choose them as the software provider. Other similar RFT terms that are being used to worldwide are Express of Interest (EOI), Request for Information (RFI), Request for Proposal RFP) or Request for Quotation (RFQ). Some of the common content in an RFT are; (1) Tender procedure, (2) Requirements and (3) Contract forms. In this publication, we will be focusing more on item no 2. Similar term that are being used to represent item no. 2 could be User Requirement Specification (URS), User Requirement, User Specification etc.

Once the RFT is being released, the next course of action will be the task of the software vendors to prepare a proposal that will then be evaluated by the software purchaser team in selecting the so-called best proposal. The main challenge of software vendor is in the proposal presentation session. They must make sure that the materials that are prepare for the session is enough to convince the panels but also must show that their solution is the best. For a software development tender, software vendors can use different approaches and methods available in the market in order to achieve their target. Based on the requirements that is included in the tender document, software vendors will be including a prototype of the requested software in order to show the evaluation panel the outcome. But at the end of the day, panels will also look for the uniqueness of the software vendor solution and the worthiness of the vendors themselves.

Meta-requirements as describe by researchers is the most basic form of requirement. Its main role in ISDT as the main artefact that describe the most basic function that a system must have, well in theory that is. The definition of ISDT is an initial draft of establish relationships between components of a system to achieve a specific result. Design theory must address the question of how to combine components and relationships to make subsystem and how to combine subsystems and relationships to make a systems [1].

The objective of this publication is to propose an initial model of formalizing USR and meta-requirement. The remaining five sections are as follows. Section 2 presents the related literature of the study. Section 3 describe briefly on a proposed model of mapping URS towards Meta-Requirement. Finally, Section 4 concludes and outlines some possible future works

2. Related Literature

This section will describe the related

2.1. Challenges Faced by Software Vendors in Request for Tender

Usually in the event of presenting software development project proposal for the evaluation panel, the use of prototype is a way to increase the panel's level of confidence towards the software vendor. Prototype are usually based on past successful project or it could an out of the wild combination of GUI and functions from a few different software projects. For a software vendor with strong background with similar past successful project, this decision of whether to take on the project or not would be a snap.

In a scenario where specifications are already included as part of the tender document that describe a detail list of thousands of specifications of the desired end-product, it is quite a daunting challenge where producing a highly convincing prototype that will meet and fulfil the specifications of the desired end-product. This is the task that will be led by Sales staff with the help and advice from a team [2] that consist but not limited to; (1) Sales staff, (2) Project Manager, (3) Technical specialists (Requirements Engineer), (4) Implementers, (5) Finance and commercial specialist, and (6) Legal experts. Some of the limitations [3]–[6] that are imposed to these team are; (1) Time. This is a case by case scenario whereby the timeframe of between the advertisement of the RFT and project proposal preparation is quite short, (2) Resources. In current situation where the is no guarantee that the software vendor will be awarded with the project, not much resources will be invested in partaking this effort and (3) Competitive. Depending on the RFT itself, the level competitiveness will vary. It is worth mentioning that even large will invest their time in small projects. Figure 1 dictates the three main target of requirements engineer.

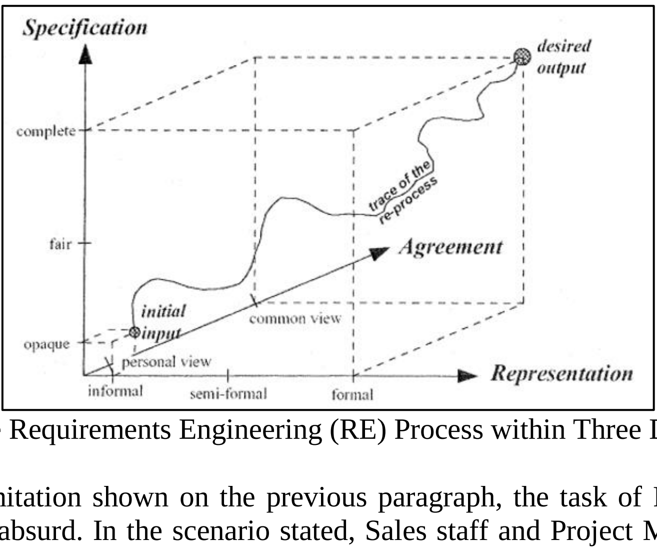

Figure 1. The Requirements Engineering (RE) Process within Three Dimension [7]

Based on the limitation shown on the previous paragraph, the task of Requirements Engineer has somewhat seemed absurd. In the scenario stated, Sales staff and Project Manager will be relying on the input from the requirement engineer to ensure that the proposal will be up-to-par with the expectation of the decision maker. Based on Fig. 1, requirements engineer main target is the achieve the desired output of a software development project by balancing the three main aspect of a software project [5]; (1) Specification, deals with the degree of requirements understanding at a given time (2) Representation, copes with the different representations (informal and formal languages, graphics, sounds etc.) used for expressing knowledge about the system. Within RE there are three categories of representations. and (3) Agreement, deals with the degree of agreement reached on a specification.

With the current attention of a Requirement Engineering is briefly illustrated in Fig. 1, it is acknowledged that with the combination of the situation of proposal presentation of a tender with the standard of work that needed to achieve by Requirement Engineers is quite challenging. Hundreds or even thousands of specifications that are included in the RFT document are also representing the almost finalize user requirement. Also, keep in mind that Requirement Engineers will need to abide with the general guideline of reviewing the quality aspect of the requirements, as shown in Table I.

Table 1. Characteristic of Good Quality Requirement [8]

No	Characteristic	Description
1	Cohesiveness	Showing the cohesiveness of a requirement and the other parts of the requirements that will be working together with that requirement
2	Completeness	The entire requirements specification should be complete and contain all relevant requirements and ancillary material depending on the standard and define template and also individual requirements should also be complete
3	Consistency	Requirements must be consistent to its source and other related requirements
4	Correctness	Individual requirements should be correct in consideration of semantics, organization business objective and requirements syntactic
5	Currency	Individual requirements must be up-to-date and not obsolete based on changes that will occur during the whole development timeline.
6	Customer/User Orientation	Individual requirements should be oriented around the needs of the customers and users if they are to be understandable and validatable
7	External Observability	Individual requirements should only specify behaviour or characteristics that are externally observable
8	Feasibility	Requirement must be able to be implemented within the scope of project budget, schedule, software, hardware and other constraints
9	Lack of Ambiguity	Requirements must be clear with no room for misinterpretation.
10	Mandatory	Questioning the nature of mandatory requirements and differentiating the requirements that is on the “wish list”
11	Metadata	Individual requirements should have that will characterizes each of them. Metadata can include but is not limited to acceptance criteria, allocation, assumptions, identification, prioritization, rationale, schedule, status, and tracing information
12	Relevance	Each requirement must be relevant and within the scope of business, application or component
13	Usability	Each requirement must be understandable and usable by stakeholders in the scope of control, evaluation, controlling cost and schedule, architectural significant, implementation and testable.
14	Validatability	Individual requirement must fulfil the need and desire of the primary stakeholders. Ensuring each requirement is accordance to the customer representatives wanted and needed.
15	Verifiability	Requirements must be verifiable against its source and associated

standards, guideline, template etc.

If the thousands of specifications that are included are being taken lightly, then software development work will projects suffer failure due to user requirements related problem such as changing requirements, unclear, ambiguous and unusable requirements and misunderstood user requirements and the failure to freeze requirements [9]–[13]. Cost and delivery overruns are resource or economic factors. Cost overruns and missed delivery can result in project termination. Publication by [14] includes increase of costs and timeline, actual project expenditures and delivery below the estimates and insufficient budget. These indirect factors may be the reason for the overrun. Failure can also be due to time and delivery below the estimates related to estimation issues in project management. Finally, the depletion of funds can result in project termination.

2.2. Meta-Requirement in Information System Design Theory (ISDT)

In looking for alternate solution of solving Requirements Engineer task of conducting earlier requirements validation of the tender specification, our view is focused on the utilization of meta-requirements. The source of meta-requirements is being heavily discuss in the area of Information System Design Theory. The breakdown of the origin of ISDT is that it originated from Design Research Science (DRS) and focused in the field of Design Theory (DT).

In 1992, a work by J. Walls, Widmeyer, & El-Sawy [1] is the first to include the phrase “Meta-requirement” in his work with relation to the field of ISDT. The future work that comes afterword in relation meta-requirement majorly in the field of ISDT, highly cited his work. Meta-requirement that was presented in his work is quite green, whereby future researchers has made improvement in the meta-requirement is being presented.

Käkölä, Koivulahti-Ojala, & Liimatainen [15]–[17] develops the product aspects of the ISDT for the class of Requirements and Release Management Systems (RRMS) with the involvement of meta-requirement. Advancement has been done thorough this work whereby the researcher introduces a different way of presenting meta-requirement. As part of the knowledge in DT, Meta-requirement could be the next big thing that is needed by Requirement Engineer as a guidance for them to do a quick check on the tender’s user specification. It acts as the main theory of how the user specification should be. With such problem faced by Requirement Engineers, this would be an alternative solution and a chance for the software vendor to be unique in the sense of presenting their proposal.

3. Formalization of User Requirement Specifications

The word requirement plays a crucial role in the life of a software developer. Requirement is the heart and soul of a system. Development of a comprehensive and highly desirable software product will be achieved if the listed requirements are fulfilled. The task of studying, creating, checking and finalizing requirements are being uphold by person who called themselves Requirements Engineer. The process starts by collecting requirements, usually originated from the system owner with the intention of producing a comprehensive documentation that will contain a list of all agreed behaviours that the system will behave based on the boundaries that are defined, best known as User Requirements Specification (URS). But this is easier said than done. Requirements usually represent itself in an unstructured form and it is highly difficult for requirements engineer to settle down with their task even during the stage of delivering the finished system to the owner. Figure 2 show the list of activities that are par-take by Requirement Engineer and its close relation towards requirement validation.

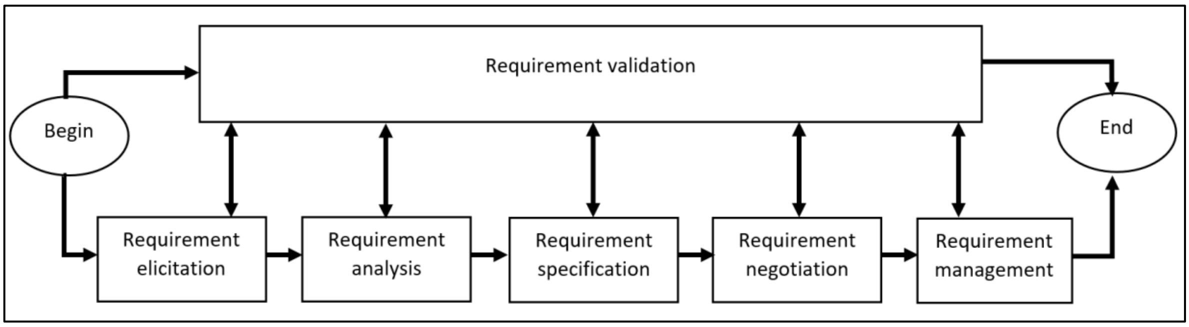

Figure 2. Requirement Engineering Relation to Requirement Validation [18]

URS document are provided earlier during the biding of tender. Different from URS that is being collect and analyse by the development team during the analysis phase, resources have already been secure through project acquisition and can be assigned during the project timeline. But for provided URS during biding of tender, normal software development organizations are not keen in investing to much resources in something that is not certain going to be theirs. The main concern of URS matter is the textual user specifications. These will be used in the meta-requirement mapping model and being considered as the crucial input of the model. Further discussion of the model will be describe in the upcoming paragraph.

3.1. Description on design theory

The main objective of introducing meta-requirements to URS is to propose an alternative solution in requirements validation. By systematically mapping the elements in both URS and meta-requirements in ISDT, the hope is to be able executing a higher abstract result of validating requirement at the earliest stages of the software project. With consideration on the consistency aspect of the URS and meta-requirements.

Based on the scope of research, it is decided that some parts of ISDT components is irrelevant. ISDT contains two parts; (1) Product hypotheses and (2) Process hypotheses. Part 2 of the ISDT component is removed from the propose meta-requirement analysis model based on the need of the model itself. Kernel Theories will also be taken out as the common ISDT will be to accommodate less than 5 kernel theories per-model. The initial model will rely mainly on the user specification product-aspect as the main input and doesn't concern on the user specification process-aspect, as shown in Figure 3.

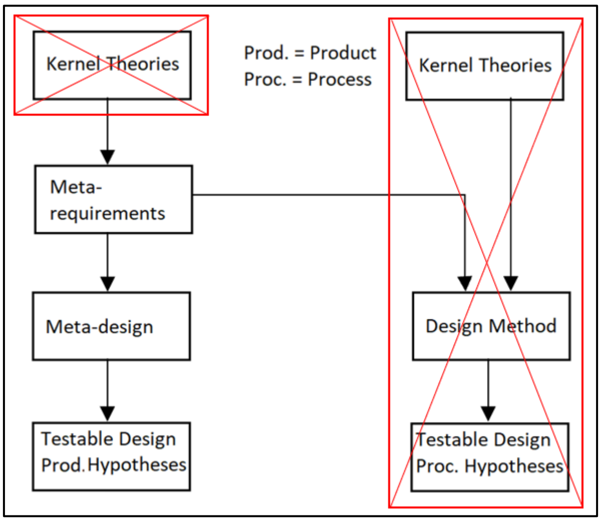

Figure 3. Excluded Components of Information System Design Theory

The remaining component of the ISDT will be carried forward in the purpose of mapping meta-requirements, meta-design and testable design product hypotheses to URS and also its requirements validation.

3.2. Mapping User specification in URS and Information System Design Theory

Based on previous paragraph, a proposed the model for the purpose of mapping meta-requirements and URS has been formulated. Figure 4**Error! Reference source not found.** illustrate the abstract level of the model. As mention in previous paragraph, the model will be focusing more on the User Specification in the USR itself. With association of "User Requirement Specification" and "Text" are generalization.

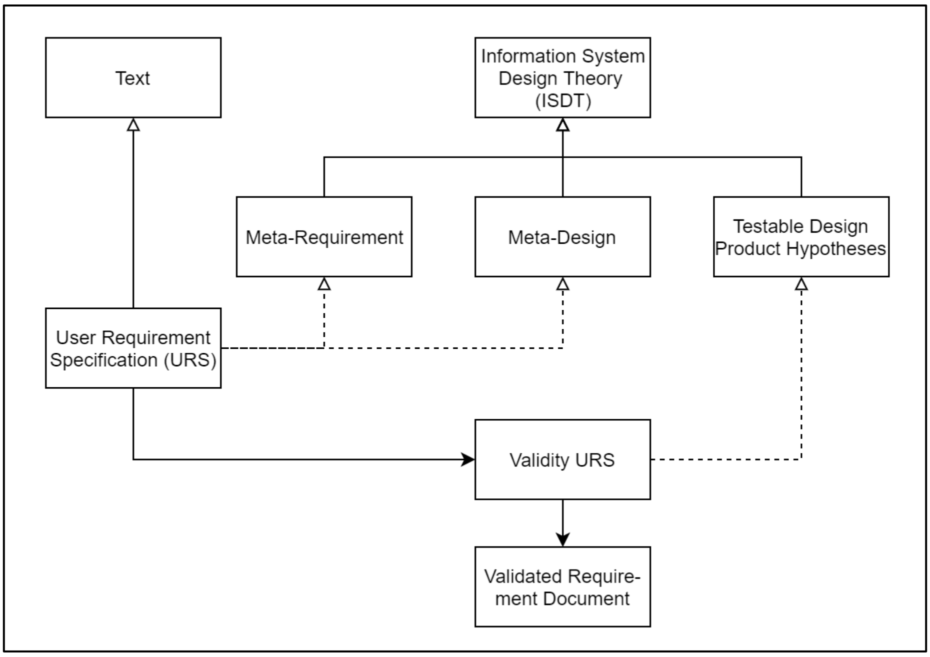
Text
Information System Design Theory (ISDT)
Meta-Requirement
Meta-Design
Testable Design Product Hypotheses
User Requirement Specification (URS)
Validity URS
Validated Require-ment Document

Figure 4. Meta-requirement and User Specification Mapping Model

Realization/Implementation of the "User Requirement Specification" towards the suitable "Meta-Requirement" and "Meta-Design" and "Validity User Requirement Specification" towards "Testable Design Product Hypotheses" will ensure that the model would able to achieve the objective of "Validated Requirement Document". The three main component of ISDT in Figure 4 will be interrelated, where a set of Meta-Requirement will be mapped to a set of Meta-Design where it could be uniquely mapped one-to-one or more. And finally, the meta-designs will also be mapped to the Testable Design Product Hypotheses in a similar fashion as Meta-Requirement to Meta-Design. This will enable the validation of URS in the sense of conformity and consistency of the URS vs the ISDT.

All of this will be done based on the development of an algorithm with the assistance of Object Constraint Language (OCL) that will support the needed mapping of the proposed model. Model will be focusing towards the measuring the consistency of the User Requirement Specification with the assistance of ISDT.

4. Discussion and conclusion

The proposed model of URS and Meta-requirement mapping is an effort towards producing an earlier validation of URS in the scope of consistency. With the current problems faced by software vendor in the event of an RFT, it is with high hope that the proposed model will become the pioneer of future research in utilizing meta-requirements and its contribution in the knowledge of requirement engineering.

Future research will be focusing more on the effort of realizing the correct method of creating those association fully executable. Keeping our eye on the horizon, the method of Object Constraint Language (OCL) is the method of choice and appropriately lean toward achieving the research objective. Further research will be conducted on the implementation of OCL with the purpose of realizing all the define associations in the proposed model will be next course of action.

References

[1] J. G. Walls, G. R. Widmeyer, and O. A. El Sawy, "Building an Information System Design Theory for Vigilant EIS," *Inf. Syst. Res.*, vol. 3, no. 1, pp. 36–59, 1992.

[2] D. Nickson, *Bids, Proposals and Tenders : Succeeding with effective writing.* British Informatics Society, 2012.

[3] Y. Lu and T. Käkölä, "An Information System Design Product Theory for integrated Order, Transportation and Warehouse Management Systems," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 3717–3726, 2013.

[4] T. Käkölä, M. Koivulahti-Ojala, and J. Liimatainen, "An Information Systems Design Theory for Integrated Requirements and Release Management Systems," 2009.

[5] M. Fatehah and V. Mezhuyev, "Design and process metamodels for modelling and verification of safety-related software applications in smart building systems," in *ACM International Conference Proceeding Series*, 2018, pp. 60–64.

[6] M. I. U. Ong, M. A. Ameedeen, and I. E. Kamarudin, "Meta-requirement method towards analyzing completeness of requirements specification," in *Advances in Intelligent Systems and Computing*, 2019, vol. 881, pp. 444–454.

[7] K. Pohl, Klaus, Pohl, and Klaus, "The three dimensions of requirements engineering: A framework and its applications," *Inf. Syst.*, vol. 19, no. 3, pp. 243–258, Apr. 1994.

[8] D. Firesmith, "Specifying good requirements," *J. Object Technol.*, vol. 2, no. 4, pp. 77–87, 2003.

[9] L. Wallace, M. Keil, and A. Rai, "Understanding software project risk: a cluster analysis," *Inf. Manag.*, vol. 42, no. 1, pp. 115–125, Dec. 2004.

[10] Y. K. Dwivedi *et al.*, "IS/IT project failures: A review of the extant literature for deriving a taxonomy of failure factors," in *IFIP Advances in Information and Communication Technology*, 2013, vol. 402, pp. 73–88.

[11] A. A. Alshazly, A. M. Elfatatry, and M. S. Abougabal, "Detecting defects in software requirements specification," *Alexandria Eng. J.*, vol. 53, no. 3, pp. 513–527, 2014.

[12] W. Al-Ahmad, K. Al-Fagih, and K. Khanfar, "A Taxonomy of an IT Project Failure: Root Causes.," *Int. Manag. ...*, vol. 5, no. 1, p. 14, 2009.

[13] R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying software project risks: An international Delphi study," *J. Manag. Inf. Syst.*, vol. 17, no. 4, pp. 5–36, Mar. 2001.

[14] K. Ewusi-Mensah, "Software Development Project Failures," in *Software Development Failures*, 2019.

[15] A. Salo and T. K. Käkölä, "Groupware support for requirements management in new product development," *J. Organ. Comput. Electron. Commer.*, vol. 15, no. 4, pp. 253–284, 2005.

[16] Y. Lu and T. Käkölä, "An information system design product theory for the abstract class of integrated requirements and delivery management systems," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 3677–3686, 2014.

[17] P. Forselius and T. Käkölä, "An information systems design product theory for software project estimation and measurement systems," *Proc. 42nd Annu. Hawaii Int. Conf. Syst. Sci. HICSS*, pp. 1–10, 2009.

[18] S. Maalem and N. Zarour, "Challenge of validation in requirements engineering," *J. Innov. Digit. Ecosyst.*, vol. 3, no. 1, pp. 15–21, Jun. 2016.

Acknowledgments

This work was supported in part by University Malaysia Pahang under Grant RDU RDU191802-2.