International Journal of Advances in Software Engineering & Research Methodology– IJSERM
Volume 1 : Issue 2
Publication Date : 25 June 2014

Tracing Requirements

[Shah, M. I]

***Abstract*—Requirements are at the core of all project deliverables. Ensuring that all project requirements are achieved by the project can be difficult to validate if no further means of tracing requirements is employed. Requirements traceability hence is a critical component of any project, as it provides the ability for a project to articulate how any given requirement has been met by the project deliverable. A great deal of research has been undertaken in this field, as requirements traceability although provides a very powerful and useful deliverable, is perceived as a very time consuming, cumbersome and labour intensive task. Due to this perception its popularity and usefulness can be compromised when short cuts are performed to minimise the time and effort that is devoted to documenting the traceability of the project requirements. This paper will look at the current state of play of requirements traceability.**

***Keywords*—requirements, traceability, techniques, challenges**

I. Introduction

Requirements as defined in [1] is a condition or capability that is required to be present in a product, service, or result to satisfy a contract or other formally imposed specification. Requirements are capabilities that a product must meet to satisfy a user's needs to solve a problem. The users' needs can come from a number of sources including compliance to a standard or to legal regulations, a business need, a business problem, market need, competition etc [2]. Requirements encompass stakeholder requirements, business requirements, user requirements, process requirements, functional requirements, non-functional requirements and solution requirements.

In order to keep track of whether requirements are being met, how they are being met and by what solution they are being met by, it is important that requirements are traced through to implementation. The concept of tracing requirements is commonly referred to as requirements traceability. Requirements traceability (RT) is the ability to describe and follow the life of a requirement, in both forwards and backwards direction [3]. There are numerous ways in which requirements can be traced. However the underlying principles remain the same. A requirement once formally documented needs to be forward and backward traced. It will be backward traced to its initiating stakeholder requirement and business need. While it will be forward traced to the solution requirement it will be addressed by. The solution requirement will then drive whether it is traced to a new or amended process or whether it will be addressed by a system change. Either way, this tracing will then need to forward trace to the test scenarios that will test that the functionality relating to the originating business requirement has been met. As discussed in [4], traceability also provides means to determine the relationships and dependencies between the software artifacts which help support some software engineering activities such as change impact analysis and software maintenance. The challenges of implementing a successful and cost-effective traceability have created a compelling research agenda that has addressed a broad range of traceability related issues, ranging from qualitative studies of traceability users in industry to very technical and quantitative studies [5].

The remainder of this paper will go as follows. Section II will introduce the requirements and the concept of requirements traceability. Section III will look at the importance of supporting requirements traceability for projects in today's organizations. Section IV review past research in this field. Section V will looks at the challenges being faced in requirements traceability at present. Section VI looks at how requirements traceability in an agile software development methodology. Section VII looks at techniques employed for tracing requirements. Section VIII introduces the concept of a requirements traceability matrix. Section IX draws the conclusion.

II. Tracing Requirements

The project landscape is continually changing and evolving to keep abreast with the fast paced nature of the way in which project deliverables need to be realized. However at the core of all projects is the need to ensure that what a project has set out to achieve has in fact been met. It sounds like a simple task. However once you overlay the project aims with the business needs, goals and objectives. These then need to be expanded into requirements, develop the processes, applications and technologies that underpin the solution to deliver these initial project aims, and then ensure you are able to trace each requirement right through to delivery. One can begin to understand how such a task is in fact anything but simple. Requirements traceability is a critical component of any rigorous software development process [5]. Requirements traceability can be defined as the ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases [6].

Shah, M I
MATHS I C
GPO BOX 1771 SYDNEY NSW 2001, AUSTRALIA

SEEK DIGITAL LIBRARY

III. Importance

Requirements traceability is one of the most important and challenging tasks in ensuring clear and concise requirements [6]. Establishing traceability on a project can be time consuming [7]. It is a painstaking task, that even with the support of emerging automated techniques, there is still a substantive amount of manual work involved [8]. For many organizations, although a particular project and software development methodology may be employed, some customization will inevitably arise. Project and software development methodologies as described in the literature may not suit a given organization's business model completely.

Hence most organizations will customize a methodology or develop a hybrid methodology which better suits their business model. This customization filters down to individual deliverables, in particular the manner in which requirements traceability is performed. Some level of tracing is always considered to be useful for all the projects to be developed [9]. Although a fairly tedious task requirements traceability is a critical deliverable of any project. Requirements traceability is an important tool that can be used by many project team members in different ways, provided the traceability is always kept up to date, as detailed below in Table I:

TABLE I
USAGE OF REQUIREMENT TRACEABILITY

Team Member	Usage
Project Manager	Able to have a clear snapshot of where in the project life cycle each business need is at currently.
Business Stakeholder	Able to view what type of requirement has addressed their business need.
Business Analyst	In a forward direction, able to see how each requirement is being met; in a backward direction, able to see what business need each requirement has addressed. Quickly highlights where there are gaps in the forward or backward direction.
Process Analyst	Able to see which requirements are met by a business process (new or amended process).
Change Analyst	Able to see how each requirement is being delivered i.e. is it a change to an existing business process, the introduction of a new business process change or delivered via a change to the software application.
Developer	Able to see which requirement is being met.
Tester	Able to see the full backward traceability of the test scenario right back to the original business need.

As highlighted in Table I, traceability can provide an important support to many project team roles. However in order to leverage this investment it is necessary to have an accurate and ready to use set of traced relations. Traceability needs to be maintained while project artifacts are evolving [7].

Three main issues identified in traceability, by a survey conducted of software projects, as documented in 10], include:

1. The necessity for extra entry data when using traceability
2. A lack of understanding on how to employ traceability
3. The lack of perceived direct benefits to the main development process

Although the information that requirements traceability can provide is greatly beneficial, it has traditionally been perceived as a 'nice to have feature' due to how laborious a task it can be. Hence it is only starting to gain popularity as more automated and streamlined approaches to requirements traceability are employed.

IV. Past Research

In 2005, CoEST, the International Center of Excellence for Software Traceability was established. It's goal is to "bring together traceability researchers and experts in the field, encourage research collaborations, assemble a body of knowledge for traceability and develop new technology to meet tracing needs" [11]. One of the primary projects the CoEST has been engaged in is the Grand Challenges of Traceability (also referred to as GCT), as documented in CGT version 2.0, the single overarching traceability challenge being faced at present, is the need to achieve ubiquitous traceability defined as "traceability which is always there, without having to think about getting it there" [5]. CGT version 2.0 also identifies the following sub-challenges:

1. Purposed
2. Cost-Effective
3. Configurable
4. Trusted
5. Scalable
6. Portable
7. Valued

The first sub-challenge is purposed. As discussed early, it is important that any traceability employed by an organization is fit for purpose and supports the needs of the project stakeholders.

The second sub-challenge is cost-effective. For an additional overhead to be added to a project, it needs to be cost-effective in order to gain support both within the project and across the organization as a whole.

The third sub-challenge is configurable. It is important that any traceability that is performed is configurable, in order for it to more easily adapt and be reusable.

The fourth sub-challenge is trusted. In order for traceability to be a standard part of any project deliverable, it is important that it is reliable and it can be trusted to clearly articulate the traceability of the requirements from beginning to end.

The fifth sub-challenge is scalable. It is important that as a project grows and by its very nature needs to support some sort of change control, the traceability of requirements also needs to be scalable in order to support changes to project scope and requirements.

The sixth sub-challenge is portable. Portability is an important aspect of traceability, as requirements need to be traced within a project, but that traceability needs to be applicable to processes and applications and be applicable

SEEK DIGITAL LIBRARY

organization wide to ensure that it remains relevant and valid, to ensure the maximum benefit is obtained from it.

The seventh sub-challenge is valued. If the main and sub-challenges faced in the requirements traceability can be overcome then by default this seventh sub-challenge will as well. It is incredibly important that any viable undertaking of requirements traceability be valued, otherwise it will simply fail to exist or be of any benefit to the project team or to the wider organization very quickly.

In [12] three case studies were undertaken and the two main lessons learned from these case studies with regards to requirements traceability, can be summarized as follows:

- The importance of reduced granularity
- The important of value-based enhancements

By employing a reduced granularity approach in requirements traceability, one is able to save time and effort and instead focus more effort on the completeness and correctness of the traceability. In this manner one is able to focus the traceability effort on high-value requirements instead of more granular requirements.

V. Challenges

Unfortunately, despite its clear criticality, numerous case studies have shown that traceability can be difficult to accomplish in practice [5]. Many industry experts agree that traceability is a valuable tool and deliverable of projects, if established and maintained up-to-date. However the perceived challenges are proving a hindrance in the ability for requirements traceability to being embraced more fully within organizations and projects as a whole.

Case studies such as the following [13] [14], have highlighted that some of the main challenges of creating and maintaining traceability is the how time-consuming, costly, arduous and error prone it is for many organizations. These challenges make it hard for organization's to embrace requirements traceability, when there are so many negatives. While other research, such as that documented in [12] have highlighted the fundamental problems with requirements traceability at present.

VI. Agile Methods

There are numerous ways to trace requirements. These will vary particularly between the development methodologies employed. An analysis of existing traceability approaches was performed in [15], and it was concluded that the approaches strongly depend on traditional development process characteristics.

The waterfall software development methodology is the traditional, robust sequential development life cycle. Its greatest strength is that it is supremely logical – think before you build, write it all down, follow a plan, and keep everything as organized as possible [16], means that requirements traceability can performed sequentially through each phase of the project life cycle. By ensuring requirements traceability is kept up-to-date, during each phase of the project life cycle, requirements traceability in a waterfall approach will not feel so cumbersome and time consuming, as the time will be performed progressively throughout the project life cycle.

While agile systems development methods emerged as a response to the inability of previous plan-driven approaches to handle rapidly changing environments [17]. Agile methods implement an iterative methodology that healthily manages change, and due to its small iterative nature of development, provides tangible deliverables more quickly than otherwise could be achieved in a waterfall approach. In agile methods requirements traceability isn't that clear cut. As detailed in [18], it is strongly recommended to implement a traceability practice in agile projects to prevent problems due to the lack of a requirements specification. Unfortunately the majority of current requirement traceability approaches are based on a complete requirements specification document, that is, from a waterfall methodology approach. Thus, the need to provide a new and specific approach for traceability in agile processes seems to be a fundamental issue [15].

In order to add traceability in agile methods there are several tracing practices as highlighted in [9], which can be used individually or combined with each other, these include

1. Trace the requirements to stakeholder
2. Trace the requirements to problem description
3. Trace the requirements to product backlog
4. Trace the requirements to sprint backlog
5. Trace the requirements to code
6. Trace the requirements to database tables
7. Trace the requirements to test
8. Documentation

Documenting the requirement traceability is a labor intensive and time consuming task. During agile methods, as the software evolves, functionality will be added, removed and changed to meet the ever-changing business needs, hence it is important that the requirement traceability is maintained up-to-date, as it can become quickly out-dated within a single iteration in an agile method.

As discussed in [19], automated techniques to recover RT links are important to save resources. As reviewed in [20], research in the last decade has shown that surprising progress can be observed with regard to how much precision and automatic support can be applied to models in the early requirements engineering process, although a complete solution is still lacking [21].

VII. Techniques

There are many various techniques that are used for requirements traceability. They all vary in the amount and differences in the information they can trace, the number of interconnections they can control between information and to the extent to which they can maintain requirements traceability when faced with changes to requirements [22].

SEEK DIGITAL LIBRARY

Below is a summary of some of the more commonly used techniques employed in organizations as summarized in [15]:

- Cross referencing schemes
- Requirement traceability matrics
- Graph-based representation
- Key phrase dependencies
- Hypertext
- Integration documents

VIII. Traceability Matrix

As discussed earlier in this paper, there are numerous techniques that can be employed to fully articulate the requirements traceability for a given project. One such way is through the use of a requirements traceability matrix (RTM).

The content of a requirements traceability matrix may include, but is not limited to linking requirements to business and client value, project objectives, complexity, estimated effort, test scenarios, owner of the requirement, their source, bounding assumptions, and their current status of progress [24]. As discussed in [23], RTM is used to control evolution of the functionality that will fulfill the project requirements, the benefits of which can be summarized as follows:

- Support for moving from a traditional to a modern testing life cycle approach
- Support to project managers to distribute work effort to ensure quality results
- Accelerate final customer acceptance
- Manage project scope
- Promote clear definition for every project requirement
- Promote project team commitment to fulfilling each requirement and hence project scope

IX. Conclusion

Requirements traceability is an important tool to employ in a project setting. As shown in this paper, requirements traceability provides numerous benefits to many project stakeholders. It provides the means by which requirements can be traced both backward and forward through the project life cycle. Traditionally however, requirements traceability has been perceived as a laborious, time-consuming task that is reluctantly undertaken by the project team. As articulated by the CoEST the main overarching challenge being faced at present in requirements traceability, is the need to achieve ubiquitous traceability, that is, traceability which is always there, without having to think about getting it there. This is particularly important when an agile method underpins the development methodology of the project. The iterative nature of agile projects makes tracing requirements an important but difficult undertaking. This paper has summarized the key challenges and issues faced in requirements traceability at present, as well as the key items to consider when addressing requirements traceability in agile methods. More research needs to be done to help organizations overcome the current challenges and fully embrace requirement traceability.

References

[1] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fifth Edition 2013 Project Management Institute Inc
[2] Kumar, V.S Effective Requirements Management
[3] Gödel, O. C. Z, Finkelstein, C.W An analysis of the requirements traceability problem, 1st International Conference on Requirements Engineering, pp. 94–101, April 1994.
[4] Anguetil, N, Kulesza, U, Mitscheke, R, Moreira, A, Royer, J.C, Rummler, A, Sousa, A A Model-Driven Traceability Framework for Software Product Lines Software and Systems Modeling 9, 4 (2010) pp 427-451
[5] Cleland-Huang, J, Czauderna, A, Dekhtyar,A, Gotel, O, Huffman Hayes, J, Keenan, E, Leach, G, Maletic, J, Pushyvanyk,D, Shin, Y, Zisman, A, Antoniol, G, Berenbach,B, Egyed, A, Maeder, P Grand Challenges, Benchmarks, and TraceLab: Developing Infrastructure for the Software Traceability Research Community TEFSE May 23 2011, Waikiki, Honolulu, HI, USA
[6] Salem, A.M A Model for Enhancing Requirements Traceability and Analysis IJACSA Vol 1 No 5 November 2010 pp 14-21
[7] Mäder, P, Gotel, O, Philippow, I Enabling Automated Traceability Maintenance Through the UpKeep of Traceability Relations
[8] Aizenbud-Reshef, N Nolan, B.T, Rubin, J Shaham-Gafni, Y. Model traceability. IBM SJ, 45(3):515–526, 2006.
[9] Jyothi, V.E, Rao, K.N Effective Implementation of Agile Practices IJCSA Vol 2 No 3 March 2011
[10] Mäder, P, Egyed, A Assessing the Effect of Requirements Traceability for Software Maintenance 2012 IEEE
[11] Center of Excellence for Software Traceability, http://www.traceabilitycenter.org, March 2008.
[12] Egyed, A, Grunbacher, P, Heindl, M, Biffl, S Value-Based Requirements Traceability: Lessons Learned Design Requirements Workshop LNBIP Vol 14 2009 pp240-257
[13] Gotel, O, Finkelstein, A. Contribution structures (requirements artifacts). In RE, pages 100–107, 1995.
[14] Ramesh, B. Jarke, M Toward reference models of requirements traceability. IEEE Trans. Software Eng. 27(1):58–93, 2001.
[15] Espinoza, A. Garbajosa, J *A study to support agile methods more effectively through traceability* Innovations Syst Softw Eng (2011) Vol 7 pp 53 – 69
[16] Deemer, P, Benefield, G, Larman, C, Vodde, B The Scrum Primer 2010
[17] Highsmith, J Agile Software Development Ecosystems, Addison-Wesley, Boston, USA 2002
[18] Cao. L, Ramesh. B (2008) Agile requirements engineering practices: an empirical study. IEEE Software 25(1):60–67
[19] Ali, N, Sharafi, Z, Guēhēneuc, Y.G, Antoniol, G An Empirical Study on Requirements Traceability Using Eye-Tracking 2012 IEEE
[20] Loniewski, G, Insfran, E, Abrahão, S. A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development
[21] Menzies, T.: Editorial: model-based requirements engineering. Requir. Eng. 8(4), 193{194 (2003)
[22] Gotel O.C.Z, Finkelstein, C.W. (1994) An analysis of the requirements traceability problem. In: Proceedings of the international conference on requirements engineering (RE). Colorado Springs. IEEE Computer Society Press, pp 94–102 37. Grünbacher P, Hofer C (2002)
[23] Marone, O Requirement Traceability, a Tool for Quality Results Proceedings of the Project Management Institute Annual Seminars & Symposium Sep 7-16 2000 Houston Texas, USA pp 359-362
[24] Misch, R Critical Success Factors for Professional Requirements Management

SEEK DIGITAL LIBRARY