

Risk Management with Enhanced Tracing of Requirements Rationale in Highly Distributed Projects

Matthias Heindl, Stefan Biffi

Institute of Software Technology and Interactive Systems

Vienna University of Technology, Favoritenstraße 9-11/188, A-1040, Vienna, Austria

{matthias.heindl; stefan.biffi}@qse.ifs.tuwien.ac.at

ABSTRACT

A recent survey with project managers of highly distributed projects at Siemens Program and Systems Engineering (PSE) brought up as main challenges: more severe communication hurdles compared to collocated teams and higher effort to communicate requirements in the team. In this paper, we address requirements tracing options to facilitate risk management with requirements clarification, collaboration, and knowledge management. We propose concepts for enhanced requirements tracing that include the rationale for requirements, related decisions, their history; and stakeholder value propositions. We sketch a cost-benefit model that helps the project manager to understand what tracing approach is worthwhile to address requirements risk in a project. The outcome lays the basis for planning empirical studies at PSE.

Categories and Subject Descriptors

D.2.1 [Software Engineering]: Requirements/Specifications – *methodologies, tools*.

General Terms

Management, Theory

Keywords

Global software engineering, risk management, requirements management, requirements traceability, knowledge management, value-based software engineering.

1. INTRODUCTION

Siemens Program and System Engineering (PSE) is an independent research and development entity within the Siemens group with more than 6,000 employees. PSE software and systems development projects cover a broad range of application domains: telecommunication and information technologies, automation and control, power, transportation, medical solutions, components, and space technology. There are several approaches

at PSE to provide sufficient competence for ensuring good project performance and problem solving:

- In the large pool of employees there are many skilled people who are able to handle challenges and problems because of their *expertise and experience*.
- PSE *Support Centres* address success-critical topics to (a) bring employees with similar problems together, and (b) support communication of experience that could help to solve a current problem. Support Centres provide training and coaching for areas like project management, configuration management, usability, and testing.
- A horizontally organized subdivision of PSE, the *Competence Base*, provides experts with line-of-business-independent technology knowledge to the other subdivisions or covers peaks in demand [16].

These approaches have worked well for handling typical challenges in collocated software development projects. However, in recent years Siemens PSE increasingly started globally distributed development projects supported by new permanent offices in Eastern Europe, Turkey, and China. Several factors have accelerated this trend towards more distributed development:

- Business advantages such as proximity to the market, the pressure to decrease the time to market (around-the-clock development made possible with teams in different time zones);
- the need for geographical flexibility to get access to new business and technology capabilities from merger-and-acquisition opportunities; and
- the formation of virtual teams and corporations based on the cost-competitive use of scarce resources in a global resource pool [13].

While only few large projects have all project participants working collocated at one site, we see major differences in handling distributed and so-called highly distributed development projects. We characterize a highly distributed project as a project with team members in two or more countries. Such projects typically exhibit the following characteristics:

- By definition not all team members can work at the same location; travel delay between the project headquarter and other locations is significant, e.g., more than 2 hours.
- Thus, there is less opportunity for flexible direct (face-to-face) communication: regular meetings occur less often than

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GSD'06, May 23, 2006, Shanghai, China.
Copyright 2006 ACM 1-59593-085-X/06/0005...\$5.00.

bi-weekly; ad-hoc meetings are very hard to achieve due to the traveling delay or high costs to establish meetings, e.g., considerably more than 1,000 Euro even for a short meeting.

- The projects are rather long-term projects (longer than a year) and their infrastructure includes configuration management and requirements management tools due to considerable staff turnover.

The goal of project managers and project participants in all projects is to deliver according to stakeholders' expectations on products and services; cost, quality, and timely delivery. However, highly distributed teams in global software development projects have to address new challenges, such as project set-up, progress control, day-to-day communication, and even management of cultural issues.

The support centre for project management at Siemens PSE commissioned a survey with project managers in highly distributed projects to identify the most important challenges coming from global software development as a first step to plan focused future support activities. The survey provided a more detailed insight into practitioners' current distribution-related problems in order to highlight the needs for improvements.

While some of the factors are highly relevant for development projects in general, the survey established their specific relevance in the context of large-scale global software development.

The survey results showed generally high relevance of requirements-related topics (see table 1). Main results of the survey were:

- Misinterpretation (32% of survey participants rated this as frequent and very important challenge) and unclear rationale (38%) of requirements are major project challenges in highly distributed projects, resulting in delays and increased effort to explain requirements;
- Both the clear and agreed statement of requirements (more than 60%) and the clear link between requirements and business objectives (36%) are major success factors, if present. However, especially in highly distributed projects it is often hard to improve on these success factors if they have not been introduced and maintained right from the start.

The project managers in the survey rated future risk management countermeasures that could come from the support centre as important for addressing these issues: methods and tools for clarifying and tracing requirements in the project context, and more efficient collaboration tool support.

In this paper, we focus on issues of requirements tracing as basis for better requirements clarification, communication, and for reaching a common understanding of requirements.

The survey findings prompted us to propose concepts for extending the standard PSE requirements traceability approach with requirements rationale and with "value traces" that help communicating relationships between business objectives, requirements, and consequently technical solution approaches.

Table 1 Requirements-related project success factors and project challenges

| Most important success factors | % of partpts. |
|---|----------------------|
| Clear and agreed statement of requirements | 60% |
| Clear link of requirements to business objectives | 36% |
| Most important project challenges | |
| Unclear rationale of requirements | 38% |
| Misinterpretation of requirements | 32% |

We developed two concepts for enhanced requirements tracing that aim at including (a) the rationale for requirements, such as related decisions and requirements history, or (b) stakeholder value propositions. A next step is to gather empirical evidence for the effect of tracing in highly distributed projects. The long-term vision is to provide tool support for trace-based requirements-centered knowledge management in highly distributed projects based on the findings of the empirical studies.

The available approaches for requirements tracing are likely to bring different costs and benefits. Thus we present an initial risk-cost-benefit model to help the project manager to assess the impact of a tracing approach on his project.

The remainder of this paper is structured as follows: Section 2 summarizes of related work on global software development and requirements traceability. Section 3 explains the research issues and solution approaches we derived from the survey results. Finally Section 4 provides a summary and suggests further work.

2. RELATED WORK

This section provides an overview on related work on global software development and requirements traceability.

2.1 Tracing in Global Software Development

There are several studies that identified a broad range of challenges in distributed projects. Some studies identified problems related to cultural diversity, process and tools [13], whereas other studies focused on collaboration and requirements-related issues, like obtaining a common understanding, analysis and negotiation of requirements in distributed projects [3][4][5].

Researchers have recognized specific skills that have to be taught to software engineering students to cope with the complexities of distribution and described their approaches and strategies for teaching global software development [5][20].

There are two ways to capture relationships between requirements and development artifacts in projects that are large and complex enough to warrant the effort of tracing: (a) by capturing requirements rationale, etc. in documents, usually coming from well-structured synchronous communication, e.g., out of a workshop, or constructed by a central keeper, and (b) access to less structured documents that emerge from asynchronous project communication such as e-mails and informal memos.

Interestingly, most research has been focused on how to support requirements understanding using synchronous means, like videoconferencing, and how to analyze the quality of such approaches. However, in addition to its immediate benefits

synchronous communication may add risks to long-term projects as such communication is typically not archived and thus not easily searchable.

Although some issues can only be settled with synchronous communication, it is usually much more costly than asynchronous communication and limits some advantages of globally distributed work. To reduce the need for costly synchronous communication and to better support collaboration in highly distributed projects, we propose to strengthen requirements tracing of asynchronous communication documents to clarify requirements by documenting and relating information that is needed to understand the background and context of a requirement.

2.2 Requirements Traceability Models

We see requirements tracing as part of project risk management as requirements communicate the system properties to the developers. If the content, importance, or context of requirements are not well understood in a project, the resulting confusion is a major cause for many follow-up risks and inefficiencies in a project that cause external problems, such as project delay, low productivity, or lower system quality.

Requirements tracing is the ability to follow the life of a requirement in a forward and backward direction [11]. Gotel and Finkelstein mention some basic techniques for requirements tracing (RT), namely cross referencing schemes [9], key phrase dependencies [14], templates, RT matrices, hypertext [15], and integration documents [17]. These techniques differ in the quantity and diversity of information they can trace between, in the number of interconnections between information they can control, and in the extent to which they can maintain requirements traces when faced with ongoing changes to requirements. There are also tools to semi-automate requirements tracing [6][18].

There are several traceability models that differ in the types of artifacts to be traced. Most projects do not use a systematic tracing approach, but leave it to individuals to perform ad hoc tracing as needed to perform development tasks, such as test planning, checking requirements for consistency, or assessing the likely impact of change requests. Ad hoc tracing has several disadvantages: e.g., experts who have a strong mental model may not need to keep explicit traces; however, later in development the memory of traces may not be sufficient and much more costly and time-consuming to obtain. Consequently, recent software engineering standards (like CMMI) demand basic systematic tracing between requirements and artifacts such as design, pieces of code, and system test cases.

Ramesh and Jarke proposed a high-end traceability model [22] that extends the basic standard (see Figure 1): in addition to the white boxes (traced in the basic standards-mandated model) the gray boxes represent artifacts that would be important for some project manager to have trace information on but are usually not traced to in practice. Reasons may be low perceived value for standard projects with a low to medium degree of distribution due to added costs for tracing and unclear actual benefits in a project compared to ad hoc tracing and basic tracing as mandated by standards.

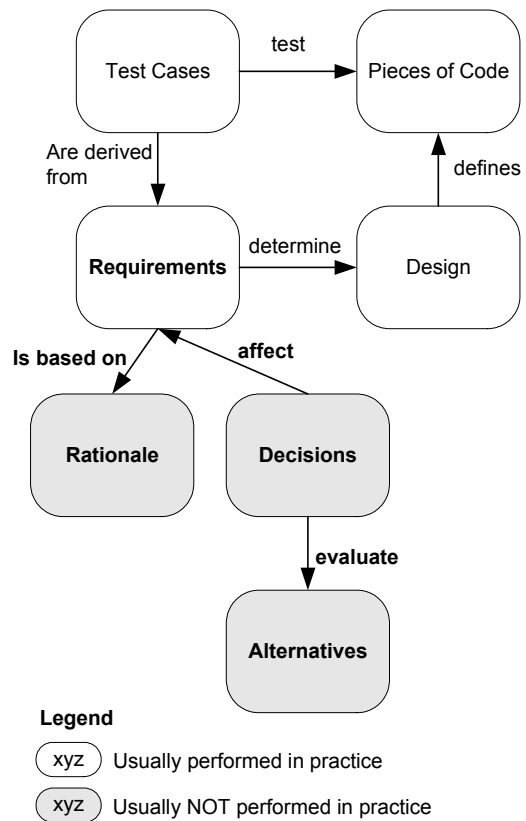


Figure 1 Extended traceability system [19].

Traces from requirements to their rationale, requirements decisions, and alternatives that were initially considered become increasingly relevant in the context of a highly distributed project as these traces provide benefits for requirements understanding in such projects that are hard to obtain otherwise, especially if informal communication in the team is costly or just does not occur.

However, the cost of tracing can be considerable. Consequently, value considerations are fundamental to motivate sustainable systematic requirements tracing activities [1]. Initial case studies and value-based software engineering analyses on systematic tracing have provided evidence on options to save considerable parts of tracing costs without sacrificing substantial benefits [7][8][12].

Requirements tracing can be seen as an investment into making the relationships between requirements and key development artifacts explicit and usable as basis for management and development activities. Thus project participants should be motivated to invest into tracing as long as they see a positive return on the invested effort. However, there are very few models supported by empirical evidence, which balance the perceived risks and the effectiveness of traces as risk counter measures, and thus would support economic trace planning. Thus project participants have a hard time to assess the likely return on tracing effort and in practice fall back to the default of ad hoc tracing.

3. RESEARCH ISSUES AND SOLUTION APPROACHES

In this paper we aim at addressing the issues raised from the survey with PSE project managers by supporting requirements clarification and understanding in highly distributed projects with appropriate traceability concepts depending on the project context and risks.

Project managers have several options for tracing: (1) non-systematic ad hoc tracing as default practice; (2) basic tracing as mandated by standards (see white boxes in Figure 1); (3) extended tracing with a schema that captures rationale for (unusual) requirements in order to support requirements clarification (see grey boxes in Figure 1); and (4) tracing requirements back to stakeholder value models in order to better understand the alignment of requirements with stakeholder value propositions (see black box in Figure 2).

Imagine an example where multiple distributed project teams that implement a web-based collaboration platform together for a customer company. A key requirement in this context is the http-server performance, which is mainly measured by the number of prospected clients requesting the server concurrently. The prospected customer, e.g., a large company that wants to improve collaboration in their distributed projects by this collaboration platform, defines the http-server performance by the following performance requirement (PR):

PR: *“The used http-server allows up to 50 concurrent client requests.”*

The requirement alone does not explain anything about its context and therefore a technical architect has multiple options to select an adequate server component, ranging from buying a new low- or high-performance server or using a server already in use at the customer site. All these options would meet the given requirement.

Documenting the context of such a requirement, e.g., by capturing the requirement’s rationale (RR) for this requirement shapes the architect’s understanding and helps to optimize his component selection decision by getting a better comprehension of the customer’s situation:

RR: *“Usually, there are not more than 50 clients in our projects that will use the designated collaboration platform.”*

For some unusual or complex requirements, these rationale bring the technical architect a better imagination of what the customer really wants.

The risk of not capturing the rationale could be that the architect made a sub-optimal component selection decision, e.g., uses a high-performance server which provides a high performance but is too expensive, while a low performance server would be sufficient.

One way to elicit requirements are requirements workshops, where the key stakeholders come together to define their requirements. Capturing the alternatives for requirements, e.g., another performance level, that were under discussion helps team members that were not joining the decision making to understand the why of a requirement. Alternatives of server selection range from low- to high-performance servers.

Modelling the stakeholder value-propositions behind requirements is an additional means to optimize decisions at the technical level. For example, the following value statement illustrates the (potential) value the desired system has for the customer:

Value statement: *“The company’s strategy is to extend the project collaboration platform to a company-wide collaboration platform after a few years, when the platform proves to be successful in projects.”*

Such value models represent the stakeholders’ interests, and the rationale for requirements can be derived from them. Finally, modeling these stakeholder value propositions and the derived rationales, results in selecting a high-performance server for the web-based collaboration platform due to the potentially very high number of prospected users after a few years. At least, the necessity for scalability of the platform was defined.

While the tracing options (1) and (2) provide support for change impact analysis and address the risk of very high additional effort to implement requirements changes in related artefacts, tracing options (3) and (4) illuminate the context of a requirement and therefore prevent wrong or sub-optimal architecture decisions, e.g., component selection decisions.

Tracing rationale and stakeholder value propositions – as additional documentation effort - seems to be especially valuable in highly-distributed projects, where development team members are distributed and need a clarifying documentation of requirements in order to perform their tasks properly. Furthermore, traceability between value models, rationale and requirements improves road mapping (customer’s strategy), trade-off decisions, and release planning, as implementation can be oriented towards these explicit value models.

In this section we provide research issues for the empirical investigation of enhanced tracing concepts and sketch a risk-based cost-benefit model that helps the project manager to understand what tracing approach is worthwhile to address requirements risk in a project. The outcome lays the basis for planning empirical studies at PSE.

3.1 Enhanced requirements tracing

The PSE survey results illustrate that there are still hurdles to take concerning requirements clarification. As face-to-face communication is hard to establish in highly distributed projects, tracing rationale could be a more efficient means to clarify requirements than currently used approaches of requirements clarification like telephone conferences.

The concept for enhanced tracing, proposed in [19], shows currently unused potential to clarify requirements. However, the cost-benefit for the enhanced traceability model (see Figure 1) needs further investigation to evaluate its benefit as countermeasure for risks stemming from unclear requirements in highly distributed projects.

Document history can explain very much of the “Why” of a current requirement version, and capturing this history, as well as related decisions and rationale for each requirement supports the understanding of a requirement. The “artifacts” we want to trace in order to support requirements clarification in highly distributed projects by tracing are:

- Rationale: explaining the “Why” of requirements, especially unusual requirements, e.g., why a certain performance is needed.
- Decisions related to requirements: history of decisions made in order to understand the history of a requirement, e.g., why a certain requirement was selected to be implemented; Especially in highly-distributed projects, which, from our point of view, are comparable to projects with high staff turnover, in terms that documenting the decisions made by others (in the past) helps to understand the status-quo.
- Alternatives representing the ways of thinking that happened before decisions in the past, e.g., which http-server options existed, and why other options were decided not to be suitable.

An interesting issue is how to capture the rationale of requirements, where it really matters. In highly distributed projects, decisions are often made by groups of stakeholders, e.g., in workshops where requirements are negotiated. As often not all stakeholders can participate in these workshops there is a need to document the workshop outcomes for project members that did not participate (and future project participants). In the workshop, some of the emerging requirements may seem clear to all participants, while others obviously warrant a more detailed description of their rationale for project members that did take part in the workshop.

To ensure the documentation of unclear requirements (a) the moderator can ask for clarification in the workshop; or (b) immediately after the workshop when a review can be conducted that raises issues early, when the memory of participants is still fresh. Once basic tracing is established, extended tracing takes little extra effort if done as soon as new requirements emerge.

In PSE practice, there is a point in time, a “requirements validation milestone”, where the requirements specification is reviewed and released. This is the latest point, where all requirements have to be clarified, e.g., by tracing rationale.

A cost-benefit model for assessing the different tracing approaches weighs the extra cost for tracing with the benefits of reduced risk and more efficient development. The costs of tracing mostly come from the effort involved in generating, reviewing, and maintaining traces to ensure their correctness over time. The benefits of tracing come (a) from the probability and impact of risks in a project if systematic traces are not available, (b) from the effectiveness of the generated traces to counter requirements-related risks, and (c) less tangible soft factors from generally more efficient development work based on tracing information. In practice the risk exposure depends on number of requirements and artifacts, on their volatility, the share of requirements that need extensive clarification. In the surveyed projects we found the following cost drivers: (1) communication effort necessary to clarify a requirement; (2) number of project participants involved in order to clarify a requirement; and (3) effort to document a requirement, e.g., in a requirements management tool.

Research issues are studies to gather empirical evidence:

- To identify types of highly distributed projects at PSE that could benefit from capturing rationale for unusual requirements;

- To identify for which requirements such additional tracing effort pays off, e.g., tracing rationale and stakeholder value models for an easy-to-understand requirement (“the background color should be green”) would be an overkill, whereas it would be crucial for more unusual requirements to reach a common understanding of all distributed project members. Furthermore, requirements exist at different levels of abstraction. Empirical analysis on which level such tracing approaches are valuable is needed.
- To perform a risk analysis to identify the risks of not documenting rationale of requirements;
- To identify processes for these project types that describe how to capture rationale for requirements and for which kinds of requirements these traces between rationale and requirements are extremely valuable;
- To gather data for cost-benefit analyses of tracing rationale in comparison to currently used approaches of requirements clarification.

Our analysis will be oriented towards Boehm’s slogan: “If it is risky not to specify a requirement, then specify it; if it is risky to specify it, then do not specify it.” [2]

Over-specifying requirements brings the risk of being too expensive, whereas under-specifying requirements, by omitting tracing rationale for unclear or unusual requirements results in expensive additional communication and clarification effort that might cause delays.

By analyzing the points above we want to find out in which projects and for which requirements tracing rationale pays off and for which not.

3.2 Linking requirements to business value

Extended tracing captures the stakeholder rationale for a requirement, which allows the developer, in principle, to understand the reason for a requirement or ask the stakeholder who originated the requirement. Another strategy links stakeholder business objectives into the traceability model in order to identify interdependencies between requirements in a project and the project’s business superstructure, as depicted in Figure 2.

The need to extend the traceability model was raised by the survey we performed at Siemens PSE, which raised need to have clear links between business objectives and requirements. This serves to set priorities for product release planning, road mapping, and to review the actual value contribution of a project at a milestone rather than just reporting resource usage and requirements fulfillment.

Similar to defining utility models for negotiation, each stakeholder can define his utility model by defining a small number of orthogonal key measures, e.g., availability, usability, which allow the stakeholder to describe the value that a system variant has for him. Tracing requirements back to these value/utility models allows better alignment of project-internal decisions, e.g., design or architecture decisions, and also release planning decisions.

These “value traces” (traces between requirements and stakeholder utility models that describe the value that a

stakeholder has from implementation of the requirement) have several benefit for project managers:

- Projects have to be able to respond to feedback and also to be able to keep pace weekly or monthly with changing business or organizational requirements. Projects must continuously monitor their relevance to stakeholders [10].
- Requirements are changing faster due to external changes [10]; the impacts need to be understood in the business context.
- Clear vision of the value contribution of a project can make a huge difference for the project outcome as it provides a common focus for rational decision making, e.g., for release planning, system design, and quality assurance planning. When people understand the overall direction, they tend to make supporting local decisions.

These traceability extensions are a means to support requirements understanding by making the value transparent throughout the project. In this context, we want to empirically evaluate how value traces can be realized. The realization contains value definition, value translation, and value controlling as proposed in value-based software engineering [1].

- Value definition (on business level): Linking requirements to business value and/or business cases. Value can be defined as set of win conditions that represent the long-term stakeholder value proposition, not just initial user needs.
- Value translation: Establishing traceability between win conditions and requirements supports understanding of the value of a single requirement. It is further work to empirically evaluate the benefit of such value traces.
- Value controlling: Unlike so-called earned-value systems, which actually track used costs and resources, value traces allow real value-based project controlling by checking the alignment of project results to current value propositions.

Traceability models extended with value traces help to address risks from changing stakeholder utility values in the project context. These changes may happen without changing requirements, but may have a major impact on project value. Thus traces between requirements and stakeholder value can support the project manager to be aware of these links and check them regularly for validity. The effort for generating and maintaining value traces is part of risk management; however, with proper support that propagates the links throughout the distributed developers, this knowledge can benefit a range of management and design decisions in the project.

4. SUMMARY AND FURTHER WORK

In this paper, we took a risk management view on requirements tracing options to facilitate risk management in order to support requirements understanding in highly distributed projects. We proposed concepts for enhanced requirements tracing that included the rationale for requirements, related decisions, their history; and stakeholder value propositions. We provided research issues for the empirical investigation of enhanced tracing concepts and sketched a risk-based cost-benefit model that can

help the project manager to understand what tracing approach is worthwhile to address requirements risk in a project.

Next step is the empirical investigation of the projects that were the basis for reporting in the PSE survey. The survey results identified risks from requirements management as a key focus for improvement. Thus, we will make a classification of the actual challenges in the different projects to compare the perceived and actual risks and the likely cost-benefit of tracing counter measures.

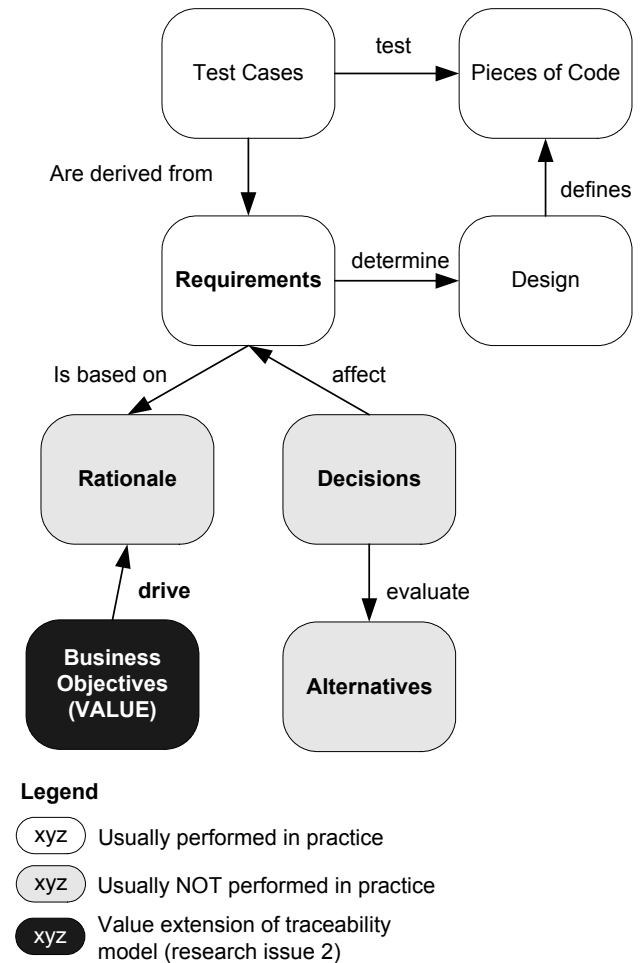


Fig. 2 Traceability Model enhanced with Stakeholder Value.

An important aspect that influences the cost and quality of tracing is tool support. We plan to perform case studies in highly distributed projects at Siemens PSE to the feasibility, effectiveness, and efficiency available tools, usually a configuration management tool, and a requirement management tool, and tool extensions to support requirements traceability.

REFERENCES

[1] Biffel S., Aurum A., Boehm B., Erdogmus H., Grünbacher P.: Value-based Software Engineering; Springer; 2005

- [2] B. Boehm, "Software risk management", IEEE Computer Society Press, 1989
- [3] Damian, D.E., Zowghi, D., Requirements Engineering challenges in multi-site software development organizations. Requirements Engineering Journal, 2003. 8: p. 149-160.
- [4] D. Damian, D. Zowghi, An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations, Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), 0-7695-1874-5/03 \$17.00 © 2002 IEEE
- [5] D. Damian, A. Hadwin, Ban Al-Ani, Instructional design and assessment strategies for teaching global software development: a framework
- [6] A. Egyed, "A Scenario-Driven Approach to Traceability", Proceedings of the 23rd International Conference on Software Engineering (ICSE), Toronto, Canada, May 2001, pp. 123-132
- [7] Alexander Egyed, Stefan Biffel, Matthias Heindl, Paul Grünbacher, "Determining the cost-quality trade-off for automated software traceability", November 2005, Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering ASE '05
- [8] Alexander Egyed, Stefan Biffel, Matthias Heindl, Paul Grünbacher, "Early traceability concepts: A value-based approach for understanding cost-benefit trade-offs during automated software traceability, November 2005, Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering TEFSE '05
- [9] M.W. Evans, "The Software Factory", John Wiley and Sons, 1989
- [10] Tom Gilb, "Competitive Engineering", Elsevier Butterworth-Heinemann, 2005
- [11] O. C. Z. Gotel, A. C. W. Finkelstein, "An analysis of the requirements traceability problem", 1st International Conference on Requirements Engineering, pp. 94-101, 1994
- [12] Matthias Heindl, Stefan Biffel, "Requirements: A case study on value-based requirements tracing", September 2005 Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-13
- [13] Herbsleb, J., Paulish, D., Bass, M., Global Software Development at Siemens: Experience from Nine Projects, ICSE'05, May 15–21, 2005, St. Louis, Missouri, USA.
- [14] J. Jackson, "A Keyphrase Based Traceability Scheme", IEE Colloquium on Tools and Techniques for Maintaining Traceability during Design, 1991, pp.2-1-2/4
- [15] H. Kaindl, "The Missing Link in Requirements Engineering", ACM SigSoft Software Engineering Notes, vol. 18, no. 2, pp. 30-39, 1993
- [16] Stefan Lasser, Michael Heiss, "Collaboration Maturity and the Offshoring Cost Barrier: The Trade-Off between Flexibility in Team Composition and Cross-Site Communication Effort in Geographically Distributed Development Projects", Proceedings of the IEEE International Professional Communication Conference (IPCC 2005), Limerick, Ireland, 10-13 July 2005, Thread: Engineering Management, pp. 718-728
- [17] M. Lefering, "An Incremental Integration Tool between Requirements Engineering and Programming in the Large", Proceedings of the IEEE International Symposium on Requirements Engineering, San Diego, California, Jan. 4-6, pp. 82-89, 1993
- [18] F.A.C. Pinheiro, J. A. Goguen, "An Object-Oriented Tool for Tracing Requirements". IEEE Software 13(2), 1996, 52-64.
- [19] B. Ramesh, M. Jarke, "Towards Reference Models for Requirements Traceability", IEEE Transactions on Software Engineering, Vol. 27, No.1., 2001
- [20] I. Richardson, A.E. Milewski, P. Keil, N. Mullick, Distributed Development – an Education Perspective on the Global Studio Project