A Survey on Usage Scenarios for Requirements Traceability in Practice

Elke Bouillon, Patrick Mäder and Ilka Philippow

Department of Software Systems, Ilmenau Technical University
Ilmenau, Germany
{elke.bouillon|patrick.maeder|ilka.philippow}@tu-ilmenau.de

Abstract. [Context and motivation] Requirements traceability is known as an important part of development projects. Studies showed that traceability is applied in practice, but insufficient tool- and method-support hinders its practical use. **[Question/problem]** We conducted a survey to understand which traceability usage scenarios are most relevant for practitioners. Gaining this information is a required step for providing better traceability support to practitioners. **[Principal ideas/results]** We identified a list of 29 regularly cited usage scenarios and asked practitioners to assess the frequency of use for each in a typical development project. Our analysis is restricted to those 56 participants that were actively using traceability in order to ensure comparable results. Subjects held various roles in the development and reported about diverse projects. **[Contribution]** This study provides not only an initial catalog of usage scenarios and their relevance, but also provides insights on practitioner's traceability practices. In result, we found all scenarios to be used by practitioners. Participants use traceability especially for: finding origin and rationale of requirements, documenting a requirement's history, and tracking requirement or task implementation state. Furthermore, we highlight topics for ongoing evaluation and better method and tool support in the area of requirements traceability.

Keywords: requirements traceability, traceability usage, usage scenario.

1 Introduction

Requirements traceability is an important part of a software development process and defined as the "ability to follow the life of a requirement in both a backward and forward direction" [8]. Traceability influences the quality of software products positively, supports changes throughout the development life cycle and eases reuse of software assets [7, 19, 21]. The importance of traceability is also demonstrated, as it is a precondition for the development of safety-critical systems in various domains, e.g., in aerospace (ISO12207, DO-178B), and in railways (EN50128). In addition, traceability is required for a certified development process according to process standards like CMMI, SPICE and the telecom TL9000. Nevertheless, numerous authors also point out that the practical use of traceability is often hindered by problems in its implementation and application, [2, 8 16, 17]. There are five main problems: (1) high manual effort for up-to-date traceability links, (2) ad-hoc traceability without strategy, (3) insufficient tool support, (4) creator and user of links are often not identical, and (5) distributed development projects require traceability across organizational boundaries.

The community of traceability researchers agrees on the fact that project-specific traceability strategies are required for a successful traceability implementation [6, 10, 21]. However, that does not mean that each traceability strategy and their parts are unique. Our hypothesis is that parts of a strategy are reusable across different projects.

Currently, very little is known about how practitioners use traceability and what they demand from it [17]. We agree with Winkler and von Pilgrim that it is an important research task to study traceability practices in order to propose and develop traceability support that suits practical needs [21]. With traceability practice, we mean the way in which traceability is used by practitioners. As a way to overcome this problem, we propose the identification, the analysis and the definition of practical usage scenarios for requirements traceability. This paper focuses especially on the first step, the identification of usage scenarios. By usage scenario, we are referring to recurring situations in which requirements traceability is used for supporting a development activity. Throughout the paper, we are using the terms usage scenario, scenario, and activity interchangeably. We consider the following information relevant for capturing a traceability usage scenario:

- The development activity that is supported or enabled by the use of traceability
- The goal of traceability in this scenario
- Participating development artifacts
- Required traceability links, their granularity, and their properties
- Stakeholder roles participating in the scenario
- A scenario flow involving links, artifacts, and roles
- A creation and update strategy for required traces
- Relevance and usage intensity of the scenario

While part of that information is invariant across different projects, other parts are dependent on project specifics like size, domain, or applied development process. The goal is to abstract from such variations where possible and to capture alternatives where required. As a result, we propose to create a catalog of traceability usage scenarios. That catalog will help users, tool vendors, and researchers to develop better traceability practices, tailored automations and training material for practitioners. In this paper, we are reporting about a survey with focus on the identification of scenarios, their usage intensity, and their relation to characteristics of the project.

This paper is structured as follows. Section 2, summarizes key prior traceability studies. Section 3 states our research questions, describes the designed survey and profiles the subjects and projects involved in our study. Our findings are reported in Section 4. Section 5 discusses the results of the survey and its limitations are considered in Section 6. We draw conclusions in Section 7 and discuss future work.

2 Related Work

In the past, two larger traceability studies were conducted, analyzing the use of requirements traceability in practice.

Gotel and Finkelstein [8] reported in 1992 about a study with the aim of finding an explanation for the weak application of requirements traceability in practice, despite its numerous scientifically cited benefits. Major findings were that support for pre-requirements traceability should be improved and that problems existed especially with the process and its stakeholder. They found multiple perspectives on what traceability was expected to enable and on the problems experienced, conflicts particularly evident between those parties responsible for establishing traceability and those parties using it (not always identical people). Further, they found that pre-requirements traceability, referring back to the origin or the source of a requirement was in need of more attention, so the need to integrate a wider variety of data in traceability, such as source material and the people involved in the project.

Ramesh and Jarke [15] carried out a several year study at the end of the 1990s. The authors investigated especially the impact of individual factors, such as employee motivation or organizational and technical support on the use of traceability. Within this study two reference models for requirements traceability were developed. These models meet the needs of high-end and low-end traceability users at the time of the study and show the activities that the participants wanted to be supported.

Furthermore, several smaller studies focused on the state of practice in traceability [1, 4, 11, 12]. These studies aimed to identify reasons for rare use of requirements traceability in practice and derived relevant research questions. As a result, all studies recognize a discrepancy between the extensive research in the field and its current practical application. However, recent studies confirm that traceability is a topic of large interest in development projects and that it is implemented in projects to some extend [12]. Nonetheless, authors also found that traceability is rarely used. None of the discussed studies focused on concrete usage scenarios, but instead evaluated traceability in general. We found across all studies examples of possible traceability usage scenarios and incorporated them in our survey. Additionally, von Knethen and Paech [20] discuss particular interests of individual stakeholders in traceability. Spanoudakis and Zisman [19] refer to a number of possible usage scenarios for traceability. Finally, Winkler and Pilgrim [21] also collected possible usage scenarios for traceability.

Despite of all these efforts, little is still known about the practical use of traceability in development projects [17]. In order to provide better project-specific support for requirements traceability, we need to answer the following research questions: (RQ1) What are practical applications of requirements traceability? (RQ2) How important is each usage scenario and how often is it applied?

3 Set-up of the Survey

We intended that study to be an initial attempt in identifying traceability usage scenarios and in assessing their importance. A number of empirical research methods is suitable for software engineering problems and offers different benefits and drawbacks [13, 18]. We decided to perform a survey as it facilitates high numbers of participants and allows for the recognition of trends [18]. Subjects invest only a limited amount of time and can schedule the participation according to their needs.

3.1 Objective of the Survey

Our study had the goal of understanding the current state of practice in traceability usage. Specifically, we were interested in finding relevant usage scenarios and in quantitative data about how regular scenarios are applied in development projects.

3.2 Collecting Scenario Candidates

In order to create a comprehensive list of usage scenario candidates, we started to gather information from the three main literature surveys previously conducted on requirements traceability: Spanoudakis and Zisman [19], von Knethen and Paech [20], and Winkler and Pilgrim [21]. All provide lists of development activities supported by traceability (see Section 2). Furthermore, we analyzed a number of publications on requirements traceability and selected those that referred to applications of traceability. We found the studies discussed in the related work section very useful and also the following publications [9, 10]. We also analyzed websites, forums e.g. [3] and studied tool documentations.

We removed redundancies and consolidated similar activities where appropriate. Eventually, we identified a list of 29 activities that were mentioned as being supported by requirements traceability. We grouped these usage scenario candidates into six groups referring to typical facets of development processes: requirements engineering and management, project management, compliance demonstration, design and implementation, testing, and maintenance and evolution. We presented grouped scenarios together in order to support subjects in understanding the context of scenarios.

1) Requirements Engineering and Management	Source
a) Finding origin and rationale of requirements, i.e., pre-requirements traceability to regulatory and other source of a requirement	[1, 8, 9, 10, 11, 19, 21]
b) Refinement and detailing requirements	[10, 11, 15, 19]
c) Documenting a requirement's history, i.e., to be able to trace to previous versions of a requirement in order to find out about changes	[15, 19]
d) Identifying stakeholders for the ongoing development of the requirements	[1, 8, 11, 15]
e) Quality- and maturity-analysis of requirements	[10, 11, 15, 20]
f) Impact analysis, which other stakeholders are important by a change to a requirement	[19]
2) Project Management	
a) Tracking the state of requirement or task implementation in detail	[9, 11, 12, 20, 21]
b) Initial Release planning	[10, 21]
c) Progress assessment on project or subproject level for getting an overview of already implemented requirements	[9, 10, 20]

d) Task assignment to stakeholders, e.g., assignment of a requirement to a developer for implementation	Tool documentation
e) Notification of stakeholders about changes, e.g., after a change to a requirement all owners of dependent artifacts are automatically informed	Tool documentation
f) Adjusting project and release plan, e.g., in case of time limit exceeding	Suggested by pretester
3) Compliance Demonstration	
a) Analyzing requirements coverage in source code, e.g., for the customer	[9, 10, 12, 15, 19, 20, 21]
b) Traceability documentation for certification purposes	[9, 10, 11, 15, 21]
c) Justification of all written code based on specification for certification purposes	[10, 20, 21]
4) Design and Implementation	
a) Navigation between specification, design, test, and code via traces	[12, 21]
b) Navigation within artifacts of the same type, e.g., within source code	[12]
c) Design assessment based on traceability metrics, e.g., to find components that contain too much functionality and should be split	[19, 21]
d) Understanding of software artifacts, e.g., project familiarization of development team members	[15, 20]
5) Testing	
a) Development of test cases based on requirements	[1, 10, 11, 15, 20, 21]
b) Defect location within the source code for failed test cases	[11, 21]
c) Discovering regression tests to be executed after code change	[10, 15, 20]
d) Test coverage analysis of specification and code	[1, 10, 11, 15, 21]
e) Stakeholder identification for understanding behavior and solving complicated problems	[21]
6) Maintenance and Evolution	
a) Change impact analysis to determine artifacts impacted by a feature extension	[9, 10, 11, 12, 15, 19, 20, 21]
b) Change effort estimation for feature extensions	[9, 19, 20, 21]
c) Feature location and support during change implementation via use of traces	[12]
d) Reuse of specification and code components, e.g., a feature with all its implementation	[19, 21]
e) Knowledge transfer to the maintenance team, e.g., in cases where a team performs maintenance that does not include any of to original team members	[11, 12, 15, 21]

3.3 Implementation of the Survey

We aimed to address and attract participants with a variety of different roles and scopes in a typical development process. Therefore, we focused on a clear structure and generally understandable language within the questionnaire. The number of questions was reduced to a minimum in order to not annoy participants. Where appropriate, questions were complemented with a text field for capturing feedback and for capturing usage scenarios not covered in our list. The survey was implemented with Unipark EFS Survey. The language of the questionnaire was German [5].

The main part of the questionnaire referred to the 29 identified usage scenario candidates and evaluated their frequency of application within the whole project a subject reported about. Evaluating the concrete frequency in which traceability is used to support a certain development activity is difficult. Projects, developments procedures and the perception of participants vary a lot. Activities may be performed daily, others only when certain milestones are reached. Accordingly, we asked subjects to estimate the frequency in which a particular scenario is used to support a given development activity as one of the three categories: regularly, occasionally, or never. We explained that regularly should be chosen, if traceability is applied more than half of the times that an activity is performed and occasionally otherwise.

3.4 Pretest

We performed a two-stage pretest to improve questionnaire quality. First, we asked five colleagues, familiar with the topic, to complete the questionnaire and to give suggestions for improvements. Their feedback helped us in getting a coherent structure, in correcting errors, and in improving linguistic accuracy. Second, we asked five members of the target group to complete the questionnaire and to give feedback on problems they encountered. This feedback was used in creating the final version of the questionnaire. Pretest data was not used for the actual analysis.

3.5 Participants

We defined our target group as traceability-applying stakeholders of software development projects. In order to attract participants, we advertised the study at workshops and meetings of practitioner communities focused on requirements engineering and software quality. Moreover, we used social networks and forums to advertise the study in virtual practitioner groups, relevant to the focus of the study. We asked subjects to only participate, if they had an almost complete overview of the traceability usage within the project they reported about, ensuring high-quality results. As incentive for participating in the study, we offered a report with the study's results.

Over a period of 6 weeks (October – December 2011), 369 subjects visited the initial page of the survey. 117 out of these subjects responded to one or more questions. We assume that subjects that do not use traceability often just did not respond at all. Our survey does not allow us to reason about how many participants adopt traceability in practice. However, that was not the goal of our study. We decided not to use partly answered questionnaires in order to ensure high data quality and to be able to compare usage rates of traceability scenarios between another. The resulting list includes 56 participants that filled in the questionnaire completely and that also use traceability in the project they were reporting about.

We asked participants about their role in the project they reported about. We allowed multiple answers as especially in small projects people often work in more than one role. Our participants covered the following roles: project manager (41%), requirements engineer (39%), quality manager (38%), architect (25%), developer (25%), test manager (14%), and CEO (11%). This broad range of participants with different focus on the development process allowed us to capture a variety of positions and opinions on the usage of requirements traceability.

3.6 Projects

We asked subjects to answer questions with respect to a project they typically work on and to preferably refer to their current project. Furthermore, we asked them to classify this project according to three criteria: project size in terms of team members, project duration and distribution of development sites. 23% of the participants were reporting about a small-scale project with 1 to 6 team members. 45% of the participants reported about a medium-scale project with 7 to 50 team members. Finally, 32%

of our subjects were reporting about a large-scale project with more than 50 team members. Subjects could classify the duration of their project as a short-term project that runs for less than one year (24%), as a medium-term project that runs for one to three years (46%), or as a long-term project that runs for more than three years (30%). One subject told in a verbal comment that her or his current project was under continuous development for 20 years. Furthermore, we asked subjects whether the project they were reporting about is being developed at one local site (49%), at multiple national sites (25%), or at multiple international sites (27%).

4 Results

In the first three parts of this section, we analyze general questions about the traceability in a subject's project. The fourth subsection discusses particular usage scenarios applied in these projects.

4.1 Reasons for Applying Requirements Traceability

We asked subjects about reasons for the application of traceability. Fig. 1 shows the possible reasons that we offered and the percentages of subjects that selected each. For space reasons, we cannot show all results in figures, the following text contains additional results not visualized.

A majority of 80% of the participants applies requirements traceability because of expected benefits. Given that only 2% of the subjects responded not to know the reasons for applying traceability, there is a remaining group of 18% of the participants that performs traceability because they are forced to by regulations, management orders, or the development environment. Only 36% of all participants are applying traceability purely for expected benefits. Association of given replies with another shows that around one fifth of the subjects that apply traceability because of regulations (17%), management requests (18%), or development process preconditions (23%) do not expect benefits.

We associated the reasons for applying traceability in a particular project with the size of that project and found no considerable differences (85% small-scale, 76% medium-scale, and 83% large-scale projects). Considerably more subjects reporting about large-scale projects report to perform traceability because of regulations (15% small-scale, 12% medium-scale, and 39% large-scale projects). The percentages of participants that reported to perform traceability because of management requests or development process preconditions grow with the project size (management request: 15% small-scale, 32% medium-scale projects, and 39% large-scale; development process preconditions: 23% small-scale, 36% medium-scale, and 56% large-scale).

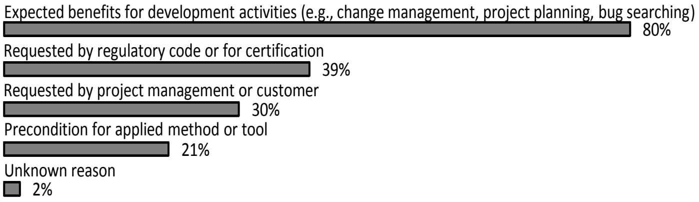

Fig. 1. Overview of replies to the question: "Why is traceability applied in your project?" (Multiple selections allowed)

4.2 General Assessment of Requirements Traceability

We asked participants to rate five statements about requirements traceability as true, partly true, or false given a subject's experience of applying traceability in the project she/he was reporting about. We allowed to rate statements as partly true in order to get an understanding of how sure subjects were in their responses. Fig. 2 summarizes responses to those questions. The figure shows that the majority of participants (63% true, 32% partly true) consider traceability as an important basis for the development process (see statement a). Also the majority of participants (48% true, 39% partly true) state that traceability should be used more actively (statement b). Nonetheless, almost the same percentage of respondents (50% true, 34% partly true) found statement c already true within their projects that the experienced benefits outweigh the cost of traceability. The cross test (statement e) supports this statement (4% true, 21% partly true). Statement d aimed to evaluate whether all development team members are involved in the traceability process, know the objectives of traceability, and also know their own role in the traceability process. Only 29% of the subjects fully agreed with that statement, 52% partly agreed. An association of those replies with project size showed that 44% of the subjects reporting about large-scale projects fully agreed with that statement, while 39% of those partly agreed. Only 20% of the subjects working in medium-scale projects fully agree with that statement, 60% partly agree. Subjects working in small-scale projects fully agree to 23% and partly agreed to 54%.

We also allowed participants to comment and report about issues in their traceability practice. More than a quarter of the participants gave such comments. The majority of those statements refer to benefits and especially costs of requirements traceability. Several subjects mention positive effects for their work or project due to traceability, but always in relation to high costs. Other subjects simply complain about too high costs and that they do not think that traceability can be applied cost efficiently in small- and medium-scale projects. Other issues that are mentioned multiple times across these comments are better tooling and better training and motivation of team members. We knew all these statements from introduction sections of research papers, but it was very enlightening to see them mentioned by practitioners struggling with the implementation and application of requirements traceability.

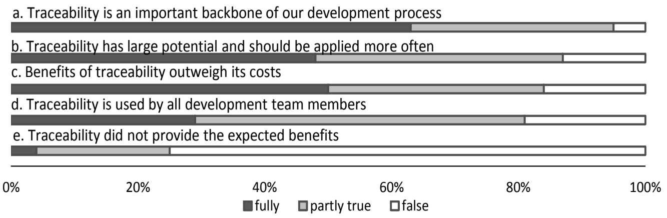

Fig. 2. Replies to the question: “Which of the following statements do you consider true, partly true, or false based on the experiences with requirements traceability in your project?”

4.3 Instrumentation of Requirements Traceability

A wide variety of general-purpose and development tools is used to instrument requirements traceability in practice. We asked our respondents about their tooling for implementing requirements traceability. We found that general-purpose applications (e.g., office, spreadsheet, and wiki) and configuration management tools were the most reported tools (both 64%). Almost equally often, subjects report about the use of integrated development environments (61%) and requirements engineering tools (57%). Only about one fourth of the subjects (27%) are using project planning and management tools for implementing traceability (see Fig. 3). A customized tooling solution is used in 39% of the projects, involving several tools that realize requirements traceability in their projects. An interesting observation is that subjects with customized solution in their project are more often fully agreeing to the statement that the benefits of traceability supersede its costs (see Fig. 2 statement c).

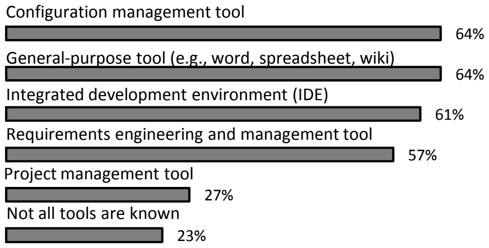

Fig. 3. Replies to the question: "Which tools for implementing and using requirements traceability are applied in your project?"

4.4 Application of Usage Scenarios

We found that the 56 traceability-using subjects in our study apply 42% of the 29 assessed usage scenarios regularly (standard deviation (sd) 22%). These subjects apply another 29% of the usage scenarios occasionally (sd 15%). The percentage of regularly used scenarios is clearly higher in large-scale projects (53%) than in medium-scale projects (36%) and small-scale projects (37%). Occasionally applied usage scenarios are less dependent on the project size (small-scale project 25%, medium-scale project 31%, and large-scale project 30%).

Table 1 list from top to bottom all 29 usage scenarios that were assessed by the subjects. The second to fourth column show for each scenario, the percentage of subjects that is performing the scenario regularly (R), occasionally (O), and never (N). The following four groups of columns evaluate the influence of project size, project duration and kind of development sites on each usage scenario. Per subgroup (e.g., small projects) we computed average usage rates and compared those against a weighted mean of all subgroups (i.e., small-, medium-, and large-scale projects). We decided for that way of comparison, as the weighted mean is independent of the number of samples per subgroup. In order to avoid a large table with figures and in order to emphasize on the interesting associations, we defined two thresholds for divergences to appear in the table. Major divergences, above 20% and below -20% of the weighted group mean, are visualized by vertical arrows (↑ and ↓), while minor divergences, in the range (10%, 20%] and (-10%, -20%], are visualized as diagonal arrows (↗ and ↘). We do not consider differences smaller or equal to 10%. They are represented as blank cells in the table. For space reasons, we only show differences for regular and occasional usage per subgroup. Differences in the percentage of subjects that never uses a scenario follow implicitly from differences in the regular and occasional usage. When reading the table, it is important to focus on the pair of R and O column per subgroup in relation with each other. For example, the arrows for the combination: scenario 1.c)–medium project are meaning that this scenario is applied 10-20% less regularly, but 10-20% more occasionally in medium projects than in the weighted mean of all projects. In the following paragraphs, we are discussing all six groups of usage scenarios and refer to the introduced percentages and influences.

1. Requirements engineering and management. All usage scenarios in this area are used in at least 70% of the reported projects either regularly or occasionally. The most regular used scenarios in this group are "traceability for finding origin and rationale of requirements" (64%) and "traceability for detailing requirements" (63%), which are even more often used in larger and longer projects. The use of traceability for "impact analysis" is with 70% combined regular and occasional usage also well established, but the low value of 25% regular use raises questions about problems in the application of traceability. We identify here a need for further detailed evaluation.

Regarding development sites, we found a high usage of traceability especially in international distributed projects. The scenarios 1.a) and 1.c) are used in all international distributed projects either regularly (67%) or occasionally (33%). We also associated the use of the scenarios in this group with the application of a specific requirements management tool. We found that the scenarios 1.a), 1.b), and 1.c) are used about 20% more regularly, if the subject was reporting that such a tool is used in the project.

2. Project management. Similar to requirements engineering and management, project management is also a popular area for the application of requirements traceability. All assessed scenarios in the group are performed in 72% of the reported projects, either regularly or occasionally. With 75%, the "tracking of requirements or task implementation progress" is the most regular performed scenario of all 29 assessed scenarios. The regular usage of traceability for this activity is even higher in national and international distributed projects.

Table 1. Average usage of all assessed usage scenarios across the projects that subjects reported about (R – regularly used, O – occasionally used, N – never used).

	Average usage rate [%]			Project size						Project duration						Dev. sites					
				Small		Med.		Large		Short		Med.		Long		Local		Nat'l		Int'l	
	R	O	N	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O	R	O
1. Requirements Engineering and Management																					
a) Finding origin and rationale of req´s	64	27	9					↗		↘				↗	↘						
b) Detailing requirements	63	18	19				↘	↗													
c) Documenting a requirement's history	54	36	10			↘	↗	↗	↘											↗	
d) Identifying stakeholders	46	32	22		↘				↗							↘		↗	↘		
e) Quality- and maturity-analysis of req´s	45	34	21							↘	↗				↘						
f) Impact analysis	25	45	30					↗													
2. Project Management																					
a) Tracking requirement/task implementation state	75	16	9			↘				↘	↗				↘			↗	↘		↗
b) Release planning	63	20	17			↘				↗	↘			↘	↗			↗	↘		↗
c) Progress assessment on project or subproject level	43	38	19		↘	↓	↑	↗	↘					↘		↘				↗	
d) Task assignment	43	29	28		↓		↗								↘			↗		↘	↗
e) Notification of stakeholders about changes	39	34	27	↘	↗			↗	↘	↓	↑	↗	↘	↗	↘		↗	↗	↘		
f) Adjusting project and release plan	34	38	28			↘				↗	↘		↗	↘				↗			↗
3. Compliance Demonstration																					
a) Analyzing requirements coverage in source code	64	25	11	↗		↘				↗				↘							
b) Pure traceability documentation	43	13	44	↘		↘	↗	↑	↘	↘										↗	
c) Justification of all written code based on specification	29	27	44		↘			↗		↘							↗		↘		
4. Design and Implementation																					
a) Navigation between specification, design, test, and code artifacts	43	32	25			↘		↗								↘		↗			
b) Navigation within artifacts of the same type	25	34	41	↘				↑		↘		↗	↗			↘		↗	↗		↘
c) Design assessment based on traceability metrics	14	45	41													↘					
d) Project familiarization of development team members	13	36	51	↘	↘		↘		↑		↘	↗									
5. Testing																					
a) Development of test cases based on req´s	61	14	25	↘				↗		↘				↗	↘	↘					
b) Defect location for failed tests	50	16	34	↘	↗			↑		↓	↗			↗	↘				↗		↘
c) Discovering regression tests	41	29	30	↘				↗		↘					↘	↘	↗				
d) Test coverage analysis of specification and code	41	25	34						↗	↓		↗							↘		
e) Stakeholder identification	36	36	28	↘			↘		↗	↘								↗			
6. Maintenance and Evolution																					
a) Change impact analysis	48	23	29									↗	↘					↗		↘	
b) Change effort estimation	39	38	23		↘	↘	↗	↗									↘				↗
c) Feature location and support during change implementation	38	30	32		↘	↘	↑	↗	↘		↘					↘					
d) Reuse of specification and code components	16	43	41	↘												↘			↘		
e) Project familiarization of maintenance team members	11	25	64														↗				

Four out of the six scenarios in this group are less regularly used in medium-scale projects. This raises a question about differences in the project management of medium-scale projects with 7 to 50 team members. We will further investigate that issue in an ongoing study. We could not find any influence of the applied tools on the use of traceability in this area. We also studied whether participants having a role in the project management of a project would favor these scenarios over others, but could not find relevant differences.

3. Compliance demonstration. Compliance demonstration is not a traditional part of a development process, but rather a major cited benefit of requirements traceability. We decided to create a separate group for scenarios with this purpose. The "demonstration of requirements coverage in code" is among the most used scenarios of all the 29 assessed. This scenario is regularly applied by 64% of the participants. In total 89% of the participants use this scenario at least occasionally. An interesting observation is that this scenario is even more regularly applied in small projects (77%).

As expected, we found strong associations between the usages of the scenarios in this group and whether one reason for applying traceability in the reported project was certification (see Section 3.1). Interestingly, the scenario focusing on the reverse activity "justifying written code based on the specification" seems far less important to the subjects. In only 29% of the reported projects this scenario is performed regularly and in another 27% occasionally. Nonetheless, we found a strong association between the regular use of this scenario and subjects mentioning certification as one reason for doing traceability in the reported project. Furthermore, we found that subjects performing this scenario also perform a large number of other scenarios in the areas of requirements engineering and management, compliance demonstration, and testing.

4. Design and Implementation. We asked subjects to assess four scenarios that are related to design and implementation of a development. These scenarios refer to navigation between artifacts, to evaluating the design of a system based on traceability, and to familiarizing the development team with the project by using traceability. The "navigation between different artifacts" is with 43% regular usage the most used scenario of this group (32% occasional usage). Only 25% of the participants use traceability regularly for navigating between artifacts of the same type (e.g. within code or within the design). We found that both navigation scenarios are more often used by larger, longer and more distributed projects. The other two scenarios seem to be less important to the subjects. Only 14% of the projects use traceability regularly for design assessment (45% occasionally) and 13% of the projects apply traceability regularly for familiarizing new team members (36% occasionally).

5. Testing. In our literature study (see Section 2) we found testing regularly cited as a software development area that is supposed to greatly benefit from established requirements traceability. The "development of test cases based on requirements" is the most regularly used scenario in this group (61%).

We found that the application of traceability for testing activities is associated with the size and the duration of the project. Smaller and shorter projects apply all scenarios less frequently, while larger and longer project apply them more frequently. A reason for this finding might be differences in the overall testing procedures in smaller and shorter projects as compared to larger and longer.

6. Maintenance and evolution. Our participants also had to assess four scenarios that were considered maintenance and evolution activities. Out of these scenarios, "change impact analysis" was the most regularly used one (48%). Overall, the three scenarios (6.a-6.c) that deal with change implementation are used in around 70% of the reported projects, but their regular usage is behind the popular scenarios supporting requirements engineering and management, project management, and compliance demonstration activities. This finding is interesting, as one should assume that traceability is especially helpful when it comes to changes and reevaluating decisions made in the past. The analysis of why practitioners often chose not to apply traceability for change related tasks is an issue for our ongoing study.

Additionally proposed scenarios. Participants had the chance to leave comments about their traceability usage and to propose scenarios that they found not covered in our list. 13 subjects used this opportunity and left comments about their traceability usage. Within these comments we found candidates for additional scenarios. Several comments referred to the application of traceability for demonstration purposes. For example, subjects want to demonstrate project progress and success to other stakeholders; they want to give rationale for why certain changes were required; and they want to demonstrate that the development followed a requested methodology. Other suggested usage scenarios are the creation of a knowledge database of project-specific decisions and issues, impact analysis of errors, and the use of traceability during release planning.

5 Discussion

Regarding our research goal (see Section 3.1), we found that usage scenarios known from literature are in fact relevant in practice and that practitioners use them. However, we also found that on average only a selection of 42% of all scenarios is applied per project. This selection usually focusses on groups of related scenarios, like requirements management or test.

Regarding the areas in which requirements traceability is most applied, we found requirements engineering, project management and compliance demonstration to be the groups with the heaviest used scenarios. We found that the most common usages for requirements traceability are: "Finding origin and rationale of requirements" (1.a), "Documenting a requirement's history" (1.c), "Tracking requirement or task implementation state" (2.a), and "Analyzing requirements coverage in source code" (3.a). These four scenarios are used in around 90% of all projects either regularly or occasionally.

Traceability is used across all types of projects, but we found that for many scenarios its usage increases with project size, project duration, and the distribution of development sites. The differences in usage that we observed are not dramatically, but clearly visible. To identify reasons for this observation, further investigation has to be done. Even more interesting are the cases, which do not follow this pattern. For example, “Analyzing requirements coverage in source code” (3.a) is more regularly used in short-term than in long-term projects.

Our results show that the usage of requirements traceability is less common in the areas design and implementation as well as in maintenance and evolution. We identify two reasons for this situation, which we aim to further explore in ongoing studies. First, later development stages deal with larger numbers of artifacts and accordingly with high numbers of traceability links to create and maintain. Second, most of the scenarios in these areas refer to tool-supported navigation between heterogeneous artifacts or build upon this ability in order to perform analyses. Our hypothesis is that more efficient and more specific tool support could make requirements traceability more attractive to stakeholders working in these areas.

The tools most used in the projects of our subjects for implementing requirements traceability were configuration management tool (64%) and general-purpose tool (64%). Along with the fact that a large number of verbal comments (50%) given by participants refer to the bad cost-benefit ratio of traceability in their projects, we hypothesize that tool-support that is better aligned to the actual usage scenarios could help in reducing cost and raising benefit of traceability. Supporting this hypothesis, we found that subjects with customized tooling perceive traceability to be more beneficial (see Section 4.3).

6 Threats to Validity

Construct validity. By conducting a questionnaire-based survey, we tried to eliminate the influence of the experimenter on the subject as far as possible. We aimed for a simple and precise language for our questions. We assigned questions assessing the usage of scenarios randomly, but grouped into phases, removing the order of questions as possible bias. A two-tier pretest was performed to gain feedback on the understandability of questions, the structure of the questionnaire, and on ambiguous or missing information in the questionnaire. The positive feedback from practitioners doing the pretest and from subjects as part of their comments suggests success in this regard. Nonetheless, due to the diverse background of our participants we cannot fully exclude misunderstandings. Our result might be biased (i.e., too positive) as participants assessed traceability usage on their own project. However, this is a general problem of online surveys and there was no incentive for them to do so. We explained in Section 3 that we decided for that research method as it allowed us to reach a larger number of possible participants.

*Internal validity***.** A problematic issue in performing a survey is finding a representative group of participants. In order to mitigate this problem, we decided to perform an online survey, which offered the opportunity for a wide population of subjects to easily participate. An often-cited barrier of online surveys, which are said to favor more technically affine subjects, seems irrelevant for a study focusing on subjects performing software engineering. We advertised our study across newsgroups and social networks. This form of advertising bared the risk of only inviting subjects that are very active and interested in the topic of requirements traceability [14]. In order to at least partly mitigate that threat, we also took great efforts to advertise our study on workshops, meetings and through personal contacts. Nonetheless, we have to acknowledge that our study mainly involved subjects that were already interested in the topic of requirements traceability.

*External validity***.** We had a relatively high number of 117 subjects that at least partly responded to our questions. Out of those we selected 56 that had fully answered the questionnaire and that applied traceability in the project they were reporting about. We were able to attract participants with a large number of roles in a project's development process (see Sections 3.5 and 3.6). This fact suggests that our data covers a variety of perspectives on the application of requirements traceability in practice. However, all our subjects were working in German-speaking companies and had mostly experiences in national and European projects. We understand our results as a trend showing the application frequency of usage scenarios within our study group. In order to gain generalizable results, more and larger studies are required.

7 Conclusions and Future Work

In this paper we reported about a survey designed to get information on how practitioners use requirements traceability in development projects. Based on a literature study, we collected 29 regularly cited usage scenarios of requirements traceability. We found that those 56 participants of our study that actually applied requirements traceability in practice use 42% of the 29 scenarios regularly. We analyzed the usage of all scenarios and found that requirements engineering and management, project management, and compliance demonstration are the areas in which traceability is heavily applied. We found that the usage of requirements traceability during design and implementation as well as during software maintenance and evolution is less common. In verbal comments, practitioners reported that they struggle with the bad cost-benefit ratio for their traceability. This problem could be addressed by more integrated method and tool support. Based on these observations, we derive two general goals for future work in the area of requirements traceability. First, goal-oriented application of traceability according to required project-specific usage scenarios. Second, developing traceability methods and tools that are adaptable to usage scenarios. For both goals it is important to understand the identified usage scenarios in more detail. We see an important task in analyzing the discovered traceability usage scenarios in practical settings in order to find out, which artifacts participate in a scenarios and what traces are required to optimally support a scenario. We are currently working on a study with a selected number of participants from this survey. In this ongoing work, we are elaborating the major traceability usage scenarios identified in this study.

Acknowledgements

We thank all participants for their help. We are funded by the German Research Foundation (DFG): Ph49/8-1.

References

1. Ahmad, A., Ghazali, M.A.: Documenting requirements traceability information for small projects. In: Multitopic Conference, INMIC 2007. IEEE International, pp. 1–5, (2007)
2. Aizenbud-Reshef, N., Nolan, B. T., Rubin, J., Shaham-Gafni, Y.: Model traceability. IBM Systems Jounal, 45(3), pp. 515–526 (2006)
3. Appleton, B: The trouble with tracing: http://www.cmcrossroads.com/agile-scm/6685-the-trouble-with-tracing-traceability-dissected (2005)
4. Arkley, P., Mason, P., Riddle, S: Enabling traceability. In: Proceedings of 1st International Workshop on Traceability in Emerging Forms of Software Engineering, pp. 61–65 (2002)
5. Bouillon E., Fragebogen: Nutzerszenarien für den Einsatz von Traceability. http://www.tu-ilmenau.de/fileadmin/media/sspi/Forschung/UmfrageFormatiert.pdf
6. Cleland-Huang, J.: Just enough requirements traceability. In: Computer Software and Applications Conference (COMPSAC '06), vol. 1, pp. 41–42 (2006)
7. Cleland-Huang, J., Gotel, O., Zisman, A.: Software and Systems Traceability. Springer, (2012)
8. Gotel, O., Finkelstein, A. C. W.: An analysis of the requirements traceability problem. In: Proceedings of the First International Conference on Requirements Engineering, Colorado Springs, CO, USA, pp. 94–101. IEEE Computer Society Press (1994)
9. Kannenberg A., Saiedian H.: Why Software Requirements Traceability Remains a Challenge. In: CrossTalk - Journal of Defense Software Engineering, July/Aug., pp.14–19 (2009)
10. Kirova, V., Kirby, N., Kothari, D., Childres G.: Effective requirements traceability: Models, tools, and practices. In: Bell Labs Technical Journal, 12 (4), pp. 143–157 (2008)
11. Klimpke L., Hildenbrand T.: Towards end-to-end traceability: Insights and implications from five case studies. In: Software Engineering Advances, ICSEA '09. Fourth International Conference on, pp. 465–470 (2009)
12. Mäder, P., Gotel, O., Philippow, I: Motivation matters in the traceability trenches. In Proceedings of 17th International Requirements Engineering Conference, pp. 143–148 (2009)
13. Padberg, F., Tichy W.F.: Empirische Methodik in der Softwaretechnik im Allgemeinen und bei der Software-Visualisierung im Besonderen. In: Gesellschaft für Informatik, Software Engineering 2007 - Beiträge zu den Workshops, pp. 211–222 (2007)
14. Punter, T., Ciolkowski, M., Freimut, B., John, I.: Conducting on-line surveys in software engineering. In: International Symposium on Empirical Software Engineering, ISESE 2003. Proceedings., pp. 80–88 (2003)
15. Ramesh, B., Jarke, M.: Toward reference models for requirements traceability. In: IEEE Trans. Softw. Eng., 27(1) pp. 58–93 (2001)
16. Ramesh, B., Stubbs, C., Powers, T., Edwards, M.: Requirements traceability: Theory and practice. In: Annals of Software Engineering, 3, pp. 397–415, (1997)
17. Schwarz, H., Ebert, J., Winter, A.: Graph-based traceability: a comprehensive approach. In: Software and Systems Modeling, 9(4), pp. 473–492 (2009)
18. Singer, J., Sim, S.E., Lethbridge T.: Software engineering data collection for field studies. In: Shull, F., Singer, J., Sjøberg, D., Guide to Advanced Empirical Software Engineering, pp. 9–34. Springer London (2008)
19. Spanoudakis, G., Zisman, A.: Software traceability: A roadmap. In: Chang S. K., Handbook of Software Engineering and Knowledge Engineering, vol. III, pp. 395–428, World Scientific Publishing Co., River Edge, NJ, (2005)
20. von Knethen, A., Paech, B.: A survey on tracing approaches in practice and research. IESE-Report, Fraunhofer Inst. Experimentelle Software Engineering, Kaiserslautern, (2002)
21. Winkler, S., von Pilgrim, J.: A survey of traceability in requirements engineering and model-driven development. Software and Systems Modeling, 9(4) pp. 529–565 (2010)