

# Deciding to Adopt Requirements Traceability in Practice

Floris Blaauboer<sup>1</sup>, Klaas Sikkel<sup>2</sup>, Mehmet N. Aydin<sup>3</sup>

<sup>1</sup>Accenture, System Integration & Technology, the Netherlands

<sup>2</sup>University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science,  
PO Box 217, 7500 AE Enschede, the Netherlands

<sup>3</sup>University of Twente, School of Management and Governance,  
PO Box 217, 7500 AE Enschede, the Netherlands

floris.blaauboer@accenture.com, {k.sikkel, m.n.aydin}@utwente.nl

**Abstract.** The use of requirements traceability for information systems development (ISD) projects is not very common in practice despite its often mentioned advantages in the literature. We conducted a case study in a large IT company to identify the factors that are relevant for the decision whether or not to adopt traceability in an ISD project. Five dominant factors emerged: development organization awareness, customer awareness, return on investment, stakeholder preferences, and process flow. It turned out that the majority of the software development project leaders we interviewed were not aware of the concept of traceability – with the obvious result that using traceability in software project is not even considered. This fact has possibly been underestimated in the present literature of requirements engineering.

**Keywords:** requirements traceability, decision-making, requirements engineering

## 1 Introduction

Requirements are a measurable statement of intent about something that a product must do; or a property that a product must have; or a constraint on a system [1]. They are the formal basis for software development. Requirements traceability refers to the ability to describe and follow the life of a requirement, in both a forward and backward direction, ideally through the whole systems life cycle [2].

Despite the fact that many scholars have studied requirements traceability from various perspectives, there is a lack of empirical studies showing if and how it is actually practiced in information systems development (ISD) projects. Most studies, as discussed later on, focus on the execution of requirements traceability and its advantages. Value added aspects of traceability are widely recognized throughout literature, and quality standards and techniques in relation to requirements traceability have been studied. However, some scholars including [3] mention that it is not a concept which is applied in every project. This was also the situation in the organization where we conducted a case study. That is, the claim about advantages

and practicing of requirements traceability has been present in the case organization, yet few projects in the research environment have actually adopted traceability explicitly into their development process. The exact reason for this has been unclear, which is the motivation of this study.

In line with this motivation, the goal of this study is to understand how practitioners go about deciding to adopt requirements traceability. More specifically, this study is aimed to identify the dominant factors influencing the decision making on adoption of requirements traceability in development projects. In contrast to other studies looking into the adoption of traceability, such as [4], [5], which look at the adoption by a development organization after this decision has been made, this study looks at the preceding stage, where the decision to apply the concept of traceability is to be explicitly made. Whereas other literature focuses on the implementation aspects and the technical aspects of the concept, this study focuses on the factors that influence the decision of applying traceability in a project from a management point of view.

For this study we adopted an explorative research approach. As an empirical analysis we have conducted a case study in a large IT development and management consulting company in the Netherlands that we'll refer to as "ITCC". Since the present literature on requirements engineering and information systems development does not provide theoretical underpinnings, accounts, models or alike, the adopted research approach is found to be appropriate to the nature of subject matter and research goal.

## 2 Traceability

There are many different definitions of traceability [5]. We follow Gotel and Finkelstein [2]: "*Traceability refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases).*" In our empirical study we speak of traceability if a conscious effort has been made to record traceability links between different products in the software development life cycle. This implies that some record has been made. It doesn't have to be with a special tool, it could be, say, in MS-Word and Excel. However, if such links exist only in the minds of software developers (conceivable on a small project), they still may be able to describe them, satisfying the definition of Gotel and Finkelstein, but we do not call that traceability.

The main value added by requirements traceability is twofold. First of all, there is the aspect of change management. Through requirements traceability, changes in the context of an application (changing requirements) can easily be analyzed for their impact on the code and test cases and vice versa, which heavily shortens the time required for software maintenance. On the other hand, increased accountability simplifies the verification of a system to its requirements and allows better monitoring of the process. However, establishing and maintaining requirements traceability is an expensive and politically sensitive endeavor. Various techniques [6] and tools [7]

have been proposed to support the realization of requirements traceability for systems development projects.

Despite these advantages, traceability is often still just an advocated desirable property of a software development process [8]. Several problems were identified in literature with regard to the implementation of traceability. A central problem is the fact that many developers see traceability as an optional activity, for which there are too few resources available and of which they see too little direct benefits [9], [1]. The reason of this limited amount of resources lies not with the developers, but should be sought one level higher, with project management. They have to release the resources required in the form of time, tooling and training [10].

Though traceability is involved with every effort in development from requirement elicitation to testing, as advocated by most ISD methods, the decision to adopt it should take place at an early stage. According to RUP and PRINCE 2, during both the startup and initiation processes of a project, the project team determines the way in which the work should be performed. The third process involved is the directive process. It is this process in which decisions on accepting or rejecting a project plan are made by a project board, whereas the contents of this plan is determined by others.

All these approaches, methods, techniques and tools proposed for requirements traceability are useful as long as its adoption decision is present preferably at the early stages of a project. However, this subject is undertheorized and calls for studying what underpins the execution of requirements traceability. That is, we need to understand how the decision on requirements traceability is made and which factors influence adoption of traceability. In the following, we present the conceptual treatment of these questions, which eventually provide us with a theoretical lens to examine this adoption in a systematic manner.

### **3 Research Framework and Approach**

In this study we adopt the viewpoint that adopting traceability is essentially a matter of choice. One can either make a choice to trace or choose not to trace during development. A decision is *a choice made from available alternatives*, which is exactly what is happening when adopting traceability or not [11]. This definition is somewhat lacking depth, however. Matheson and Howard [12] define a decision as *an irrevocable allocation of resources that is revocable only at a cost in some resource, such as time or money*.

Several different schools of decision making have arisen in the recent years. The classical decision making theory is a normative theory, which stipulates how decisions should be made on a normative basis and what the best outcome would have been [13]. This study adopts Classical Decision Making (CDM), and specifically the Stanford school of decision making of which Howard is one of the most prominent authors [14]. The theory as discussed by Howard identifies a process by which a decision is structured and should be structured for that matter. The goal of this study, however, is not to define what decision *should* be made, but *how* this decision is reached. The model is therefore not used in a normative way, but in a way of

describing the process. It is not intended to be followed to see how a decision should be made, but to identify the aspects involved with making this decision.

Howard [14] approaches the decision making process by modeling it as a shift from a real decision problem to a real action, including all the actions required to create this shift. This is a process of elicitation and analysis, leading to clarification of the actual problem in a way that it can be acted on in a logical way. Three phases are distinguished: formulate, evaluate, and appraise a decision problem, along with any actions appropriate for dealing with the problem. A decision problem goes through one or more iterations of the three phases. When the decision maker is content with the outcome of the process, meaning that he sees no need for another iteration and no further information available, he will decide how to tackle a problem and take real action.

Central in the decision making process is the decision basis. It is established during the “formulate” phase, where the required information is elicited and structured as needed. With this basis, the different solutions can be evaluated and appraised, after which, if required, another iteration of the cycle is performed. This entire cycle is based on the decision basis. The decision basis consists of three parts:

- *Choice.* There are different alternative solutions for the decision problem.
- *Information.* In order to judge the alternative solutions, information is needed. In Howard’s theory information consists of models and probability assignments.
- *Preferences.* Personal preferences do play a role in decision making. These comprise personal values, time preferences and risk preferences.

Before a problem is subject to the formulate-evaluate-appraise cycle, however, a problem must have been identified as a decision problem – otherwise there is no cycle. When the problem has been identified, an elicitation process yields input to the decision basis.

Howard’s theory was used as a structuring principle to identify possible factors influencing the decision to adopt traceability. All relevant factors for the decision to adopt traceability that were found in the literature could be subsumed by categories from Howard’s theory. The theoretical framework itself suggested some other factors. For example, the fact that awareness of traceability is needed in order to make any decision. This is self-evident, and for that reason not mentioned in the technical literature. Yet it proved to be a most relevant.

### **3.1 Relevant factors identified**

To identify existing factors that are considered as relevant, we have reviewed those studies concerned with project management, software engineering process, information systems development, and requirements engineering and structured them using Howard’s framework, as elaborated above. Here we elaborate these factors with their descriptions, rationale and the way to examine in empirical setting.

#### **Problem identification**

*Development organization awareness.* The choice whether or not to adopt traceability is made in the development organization. If this organization is not aware of the concept, it will never adopt traceability. Several roles within the development

organization can create this awareness with the decision maker (the project manager). This factor is measured by checking whether or not it was discussed with the project manager during the project's initial phases. Precondition is that those that were aware did take this effort.

*Customer Awareness.* Traceability can also be a demand of the customer. The customer should be seen in a broader sense here, not just as the manager. Acceptance criteria [15] are examples of concrete demands of the customer on traceability. This factor is measured by checking whether or not any demands with regard to traceability from the customer were known with the project manager.

### **Elicitation**

*Sources of Influence.* Literature offers little insight into the roles of members of a project team involved with the creation of the project plan. From the project board the senior user, who represents both the end-user and the IT-management department, is an important source of information required [16].

### **Decision Basis – Choice**

*Technological Possibilities.* To make sure that the alternatives are realistic, the technical support has to be adequate for the concept to be applied. For this decision, the decision maker has to be under the impression that the technical support is sufficient. This factor is measured by discovering whether or not the project manager feels the option is realistic.

### **Decision Basis – Information**

*Return on Investment.* In business, every decision can be approached as an investment decision. The costs of adopting traceability are often apparent, lying in the resources required for each registered traceability relation. Many researchers have found that one of the main problems, if not the main problem, with regard to adopting traceability in practice is the lack of perceived benefit with management, leading to a diminished support from management [3], [4]. This factor is measured by examining whether or not the project manager is aware of the advantages and drawbacks of traceability in the economic sense

*Quality Standards.* As discussed before, both CMM and ISO 9000 demand the use of traceability in development projects [17], [18]. When an organization wishes to adhere to these standards, in the case of CMM at the level where traceability is required (2 and beyond), adopting traceability is required. The organization can be both the development organization and the customer organization. This factor is measured by looking at certification of the organization.

*Compliance.* Besides quality standards such as ISO and CMM, there are also legal standards which companies have to adhere to. One of these standards is the Sarbanes-Oxley Act of 2002 which applies to all companies tradable on the NYSE [19]. This standard demands traceability from all systems in use in order to ensure transparency. This factor is measured by looking at the required adherence to legal standards in the customer organization.

*Project Complexity.* When the complexity of a project increases, it also becomes harder to comprehend the system for developers. This complexity has to do with the

familiarity of the development organization with the technology used. Traceability links between code and other deliverables aid in both the bottom-up and top-down comprehension of code by programmers, increasing their productivity through being able to both derive code from the preceding products and to place chunks of code in perspective [20]. Project complexity can be judged from high to low by the project manager.

*Development Method.* The characteristics of a method influence, amongst other things, the amount of changes that occur during the development project itself. Incremental and iterative development methods lead to more changes on already developed parts of the system during the development project, therefore increasing the value of adopting traceability. Besides simply adding to the total increase of value, the fact that these benefits lie in the development stage also improve the likeliness of the adoption.

*Product Life Expectancy.* The main financial benefit lies in the heavy reduction of the time required for impact analysis when changes occur. The longer that a system is in use, the higher the number of changes that occur during its lifespan, therefore again leading to a higher return on investment. The expected lifespan of the system is often stated at the start of a project.

*Dynamics of the Environment.* The frequency of risks is determined by the amount of changes in its environment. These changes in the environment can consist of changing interfaces, changes in business process layouts, changes in organizational structures, and many other matters. The more dynamic the environment of a system, the higher the amount of changes which is to be expected and the higher the expected return on investment for adopting traceability. This factor is measured based on the project manager's perception of it.

### **Decision Basis – Preferences**

*Stakeholder Preferences.* The outcome of a project influences many different parties. Each of these parties might have an influence on the outcome of the decision, which they would want to turn in their own favor. The manager of an IT department might be bound by a tight budget, which means that the solution has to be within that budget, whereas adopting traceability might mean exceeding the budget. This way, what is best for the customer organization is not necessarily what is best for individuals involved. Daft defines this as politics, where in this context power is exercised to influence the decision to strive for a self-serving purpose [11].

*Process Flow.* Adopting traceability increases the workload for those involved with it. It creates additional tasks which are to be performed during the work in development, which are often seen as extra and optional [21]. This factor is measured by discovering the attitude of the project manager towards the influence of traceability on the development process.

*Benefits Outside Project Scope.* The profit of traceability comes along when the links are used in change processes that occur later on in the systems life. The benefits lie not with the development project. Possibly a main driver for using traceability is the customer, in this case the IT management department. First of all, this information has to be made available. Second, the development organization has to be willing to supply what the other party wants, when the investment is of limited benefit to the development organization [3, 4].

### 3.2 The Conduct of Case Study

The case study took place in 2006 at ITCC, an IT consulting company in the Netherlands with about 4600 employees and a yearly turnover of around M€ 450. Within the requirements business unit of ITCC, one of the current topics is the professionalization of the service provided and especially the professionalization of the requirements process. One of the issues that has arisen, and not just in the requirements unit, is the notion that changes within a system can be performed a lot faster, and therefore less expensive, when the entire process is performed in a repeatable, but more specifically, traceable, way. So, the need for the conduct of this study was immediate for the organization, and the researchers got access to whatever data sources they needed.

The unit of analysis in this case study is the person who is responsible for decision making for requirements traceability. At ITCC this is the project manager (we did verify this during the case study). There are project managers with different ranks and with different levels of experience. To get a good sample of the project managers at ITCC, first we have selected eight managers with experience with different kinds of projects. It turned out that only three of them knew about requirements traceability. In some cases traceability techniques were used, but the interviewee was not familiar with the term “traceability”. So we operationalized “being aware of traceability” as recognizing it after our concept of traceability (as formulated in the first paragraph of section 2) had been explained.

Awareness about traceability is a precondition for making decision about it. Hence, we expanded the selection and conducted in-depth interviews with in total six managers who are already aware of traceability. Five of them adopted traceability in their projects, but one of them had never adopted it in any project he had been in charge of. Table 1 summarizes their profiles. In addition, interviews were conducted with the five managers in the first sample who had no knowledge of traceability and therefore did not make any decision with regard to its adoption. The interviews with these professionals were a valuable addition, due to their estimated impact of traceability on their work.

The data collection started by performing a pilot interview, as suggested by [22]. A random project manager was selected (with the known prerequisites) and the interview was performed with a focus that lied more on the process than the product. Only minor details were adapted in the interview protocol after the pilot, hence we considered it OK to keep the pilot in the data set. The interviews took place at various locations; half of them were at customer locations and the other half at ITCC headquarters. The data from the interview protocols are summarized in Table 2. The 6 managers with awareness about traceability are referred to as “respondents”; they were interviewed about a project they were currently involved in. The 5 other managers were asked for their opinion after the concept was explained. They are referred to as “informants”.

Two factors, contract type and project size, were not on our list in Section 3.2 but came up in the interviews as being relevant.

**Table 1.** The profiles of respondents

Respondent	I	II	III	IV	V	VI
Experience as PM	10 Years	8 Years	10 Years	8 Years	15 Years	2 Years
Experience with ISD	24 Years	21 Years	17 Years	8 Years	23 Years	12 Years
Work Experience	technical automation	ISD projects	7 years IT-consultant	IT strategy	Developer, Designer, Analyst	Functional Designer, Team Leader, Architect
Product	Forum Application	Call Centre	MIS	ERP system	1st Line Support System	Batch Conversion System
Project Length	6 Months	4 Months	9 Months	18 months	24 Months	8 Months
Team Size	3	12	17	6	24	14
Traceability Applied?	Yes	No	Yes	Yes	Yes	Yes

**Table 2.** The opinions of respondents and informants on the relevance of factors

Factor	Respondent						Informant				
	I	II	III	IV	V	VI	I	II	III	IV	VI
Development Organization Awareness	E	E	E	E	E	E	D	D	D	D	D
Customer Demand	+	-	-	-	-	-	+	N/A	+	N/A	+
Technological Possibilities	E	E	E	0	E	E	+	N/A	N/A	N/A	N/A
Return on Investment	+	-	0	+	+	+	+	+	+	+	N/A
Quality Standards	E	-	0	0	0	0	N/A	N/A	N/A	+	N/A
Compliance	0	0	0	0	0	0	N/A	N/A	N/A	N/A	N/A
Project Complexity	0	0	0	0	0	0	N/A	N/A	N/A	N/A	0
Development Method	0	-	0	0	0	0	0	N/A	+	N/A	N/A
Product Life Expectancy	0	0	0	0	+	0	N/A	N/A	N/A	N/A	+
Dynamics of the Environment	0	0	0	0	+	0	N/A	N/A	N/A	N/A	+
Stakeholder Preferences	0	-	+	-	0	0	N/A	-	N/A	0	-
Process Flow	E	-	E	0	E	E	0	N/A	+	N/A	-
Benefits Outside Project Scope	-	-	0	-	-	-	-	-	0	-	-
Time Pressure	N/A	-	0	-	-	0	N/A	-	-	+	-
Contract Type	N/A	-	+	-	0	0	N/A	-	-	N/A	-

Respondents: those who are aware of traceability; Informants: those who are not aware

- + : this factor positively influenced the decision to adopt traceability
- : this factor negatively influenced the decision to adopt traceability
- 0 : this factor did not weigh in on the decision to adopt traceability
- E : this factor was seen as an enabler for the adoption of traceability
- D : this factor was a disabler for the adaptation of traceability
- N/A : this factor is not applicable to the discussed project

## 4 Findings and Discussion

Based on the conducted in-depth interviews we have reorganized the factors as shown in Table 3. Obviously there are dependencies between the different factors. We identified five dominant factors and clustered the other ones with the factor they are most related to, yielding the 5 categories shown in Table 3. The five dominant factors are: awareness of the development organization; awareness of the customer organization; the perceived return on investment by the project manager; personal preferences of the stakeholders; the way traceability influences the process during the development project. We will discuss each factor and its empirical justification.

**Table 3** Classification of factors for adopting requirements traceability (dominant factors in italics).

<b>Development Organization</b>
<i>Development Organization Awareness</i>
Sources of Influence
Quality Standards
<b>Customer Organization</b>
<i>Customer Awareness</i>
Quality Standards
Compliance
<b>Financial factors</b>
<i>Return on Investment</i>
Dynamics of the Environment
Project Size
Product Life Expectancy
<b>Political factors</b>
<i>Stakeholder preferences</i>
Contract Type
Benefits Outside Project Scope
<b>Operational consequences</b>
<i>Process flow</i>
Technological Possibilities
Time Pressure

### Development Organization

*Development Organization Awareness.* Awareness of traceability in the development organization was identified as a prerequisite for the existence of the decision to adopt traceability. Where this factor might seem trivial, its validation showed not only that it was definitely an influencing factor but also that it was a quite important factor. From the first eight interviews that were planned, five interviewees had no knowledge about traceability as a concept that could be applied in systems development. Although it is not possible to draw any quantitative conclusions from this study on how many project managers do and do not know what traceability is, awareness of project managers is quite an important factor because without this awareness it is impossible to make this decision. In the five interviews with informants (i.e. those

interviewees not aware of traceability) we could see that they all show development organization awareness as a disabling factor.

In only one of the cases did anybody ever discuss traceability with the project manager before the development started. This was an architect. The fact that the informants did not know anything of the concept also shows that nobody ever bothered them with it, although this might seem like a logical move. Parts of the development organization might be aware of the existence of traceability and its advantages or drawbacks, but these were never communicated to the relevant roles in the project except for the case of one respondent.

*Sources of Influence within the development organization.* In half of the projects mentioned did anyone other than the project manager have an influence on the decision to adopt traceability. Informants only mentioned the program manager as a potential influence once, whereas he was an influence in one of the cases, but the architect was seen as a large potential influence.

*Quality Standards at the development organization.* It was often found that (unofficial) quality standards did not demand traceability which meant that this negatively influenced the decision; it did not have to be adopted. Organizations which do work through certification such as CMM at level 2 or higher would have this influencing the decision, since traceability is a must in this case. This was not found in any of the cases, but is indispensable for organizations which have regular audits on their CMM level.

### **Customer Organization**

*Customer Awareness.* Besides the influence and knowledge of the development organization, the customer also has quite an important influence on the decision to adopt traceability. Almost all project managers stated that if a customer demanded traceability, he would adopt it. However, very few customers actually demanded traceability. Whereas the IT management organization of the customer, be this an in-house department or with outsourced IT management, would most likely be the party demanding traceability, this party is often not involved in the initiating phases of a project. Few of the interviewees had contact with this party, and if acceptance criteria were given, they almost never included traceability. The interviewees regarded customer awareness as a causal factor. If the customer demanded traceability, he would perform this (as long as any required budget for this was created). On the other hand, if the customer did not demand it, it made things easier not to adopt it, enforcing the negative decision on this subject. The latter was mostly the case in this study, as can be seen in Table 2.

*Quality Standards at the customer organization.* Only one case was found where the respondent was subjected to an ISO 9000 certified process, but this did not demand traceability. If traceability was performed however, there were strict guidelines with regard to review processes which influenced traceability in a negative way.

*Compliance.* What applies to quality standards is somewhat similar to what applies to compliance. Regulations and laws such as the Sarbanes-Oxley Act and Basel II demand from organizations that they develop their systems in a traceable way. In the empirical study, no cases were found where this was demanded yet, although this would be expected in projects in the financial sector. In this case, it would become

part of the customer demand: not as a demand from the IT management department, but from the risk management department as one interviewee remarked.

### **Financial factors**

*Return on Investment.* As stated in the CDM theories, the decision to adopt traceability could in practice be approached as an investment decision, and this is exactly what was found in practice. Although not all respondents performed a quantitative analysis of the costs and benefits involved with adopting traceability, they did weigh the pros and cons in a financial sense. This happened mainly because, in the end, they were responsible for finishing the project within budget. This led to the question whether or not adopting traceability would aid in reaching this goal in several of the cases, in which it was answered as true in most cases. What some project managers, in this case two informants, showed as missing in their toolkit was being able to determine whether or not traceability was worth performing at all, especially in their project. The perceived financial benefits of traceability were different for each case. Often, the verification of traceability was seen as a lot more important than the main benefit mentioned in theory, being the simplified impact analysis. Steering the project with better management information, both with respect to the development progress and the correctness of the product was seen as the main financial advantage by one respondent, whereas one person also saw the reduction of tests required as an advantage and only one person performed traceability mainly for the impact analysis advantages.

*Dynamics of the Environment.* As stated, in only one case did the project manager identify impact analysis as an advantage of traceability. The others did not take the dynamics of the environment, and the correlated amount of changes, into account. One respondent considered the environment not very dynamic, hence it didn't have much weight; another was involved in a project where the overriding concern was to get out a system, irrespective of quality, in a very short time frame. This factor is related to 'benefits outside the project scope' discussed below.

*Project Size.* A factor that was not on our list, but did come forward in several interviews, is that of project size. Many of the project managers who were acquainted with the concept of traceability stated that they applied it mainly in larger projects. The longer a project lasts, the larger the benefits. Where the factor "product life expectancy" looks at the life of the product, the life expectancy of the project also matters with regard to the amount of changes found. For a project manager, larger projects are harder to keep track of, to keep a good overview. Traceable development offers more management information. When looking at the adoption of traceability by the respondents, there was also a trend which could be found with regard to the size of projects. Traceability was seldom adopted in small projects, but in larger projects this percentage got a lot higher. This study looked into the factor of project complexity, which focused on the technology used and the familiarity with this. Project size also contributes to complexity. The interviews indicated that size, rather than complexity, is recognized as a factor.

*Product Life Expectancy.* Only one respondent was aware of taking the expected period of use of the system under development into account with regard to the decision on adopting traceability. He felt this weighed in the cost/benefit analysis, which is also what the theory prescribed. Whereas the other respondents did perform a

cost/benefit analysis, they often did not see the advantage in the form of the improved impact analysis, so this factor did not apply to them. When one were to recognize this as an advantage of traceability the informants felt that life expectancy of the product would matter. On the other hand, the life expectancy has to do with matters which occur outside of the scope of the project, which is limited to just the development of the system.

### **Political factors**

*Stakeholder Preferences.* The different stakeholders involved in the decision could have personal motives with respect to the outcome of the decision which weighed in, and several of these were identified. In many of the interviews, budgets came forward as very important issues. The form of contract available was something discussed very often. If a project manager remains further within budget, this provides him a larger bonus after the project is finished. For many project managers, this was an influence on the decision to adopt traceability, especially in the case where traceability was not adopted. This factor turned out to be quite an important influence in the different interviews. It is closely related to the view the respondent has on the return on investment of traceability however, since for project managers it is often all about the money, the final number is what counts. If he feels traceability will not lead to many extra costs, this factor will not necessarily be of influence, whereas it becomes quite important if he feels there are additional costs involved.

*Contract Type.* The type of contract, to our knowledge, isn't mentioned anywhere in the technical literature as a relevant factor. Yet this was identified by several interviewees as quite important. Following Lauesen [23], we identify two types of contract for software development projects: fixed price and time-material based. The first type is based on a quotation provided by the supplier. This can be a response to a tender, where a request for proposal was sent out by the customer, or because the customer directly approached the supplier. A fixed price is determined for the project, any additional costs are for the supplier, but any savings with respect to the agreed price are also for the supplier. In the second type of contract, a registration of hours is kept and the price paid by the customer is determined by the actual time and materials used in the development project. What is important here, however, is the decision maker's impression of the return on investment of traceability in the development project itself. If he feels traceability will add costs to a fixed price development project, this will negatively influence his decision.

*Benefits Outside Project Scope.* A factor which was not necessarily quite clear in theory became all the more apparent in practice. The scope of a project, as defined by PRINCE2, is limited to the development and does not involve the use of the product [16]. The satisfaction of the customer is only checked at the completion of the project, the project manager is generally not responsible for any aftercare. As one of the interviewees remarked, this is quite important commercially however. A customer who remains satisfied, even after the delivery of the product, is a customer who returns to the company. Many of the project managers did not automatically think of traceability as something that had an improved impact analysis as the main advantage. For them, the fact that most of the benefits generally lie in the IT maintenance phase was irrelevant, they were very clear that all that mattered to them was the project itself and not anything beyond this. Four out of five informants also felt that any benefits

obtained beyond from the development project itself were irrelevant and would not be taken along in the decision, therefore negatively influencing this decision.

### **Operational consequences**

*Process Flow.* The flow of the process was definitely regarded as an influencing factor by the different project managers. In one case the project manager perceived the adoption of traceability as a negative influence on the process, traceability making the operational processes more complicated. In most other cases, however, the contrary was found. The project managers did not perceive the adoption of traceability as a problem with respect to the flow of the process. In the end, this factor again has to do with the perception of the decision maker of its impact and can only be influenced by experience and demonstration.

*Technological Possibilities.* Technology that supports traceability is something that is quite important to the project managers. If there is insufficient support for traceability in the opinion of the decision maker, this would influence the decision in a negative way. Nearly all of the respondents felt, however, that the technical support of traceability is quite adequate. For this reason, this positively influenced their decision to perform traceability, although it was not really a decisive argument. Technological possibilities were more seen as an enabler of the process. The use of text processors and spreadsheets was always seen as enough support, although tools were deemed as interesting by several respondents. In the case of the adoption of tools the issue of return on investment would become a lot bigger though according to them, a more thorough cost/benefit analysis would have to be performed.

*Time Pressure.* A very clear issue which came forward during one of the first interviews was time pressure. In this specific case, the project had a very strict deadline and the project manager had the feeling that implementing traceability would increase the time required to develop the system. This last thing is quite important; again it has to do with the perception of the project manager on how traceability would influence the project. If time pressure is an issue in a project, it therefore leads to a negative influence on the adoption of traceability. Many informants also stated that traceability would probably be one of the first things that were dropped as soon as time pressure became an issue and a small form of panic appeared. "Quality issues" such as documentation and traceability are seen as a lot less important than functional development, which is critical to the project.

### **Discarded Factors**

Two factors identified from the literature were not found to be relevant by the interviewees and therefore discarded from the above list.

*Project Complexity.* The interviewed project managers could not relate to this factor. Note, however, that they did mention project size, as discussed above. An explanation for not perceiving technical complexity as a factor could be that the project manager operates at a higher level of abstraction. According to Antoniol [20], it does play a role for those performing the operational development work.

*Development Method.* This factor has not played a significant role in the practice examined. Although it did influence the decision once, and one informant felt it might influence his decision. This factor was seen as hardly relevant.

## 4 Conclusions

In this study we have investigated how practitioners (project managers) go about making a decision on adopting requirements traceability. From a literature review in requirements engineering, software engineering and information systems development and Howard's theory of Classical Decision Making we identified factors relevant for making a decision about traceability. We validated these factors in a case study conducted in a large software development and management consulting company. As most important factors, in the view of project managers, we found development organization awareness, customer awareness, return on investment, stakeholder preferences, and process flow. Some factors from the literature (development method, project complexity) were not considered important, while some factors not mentioned in the literature (contract type, time pressure) were important for project leaders for making a decision about requirements traceability.

In this company, the majority of software development project leaders are not aware of the concept of traceability – hence using traceability is not even considered. This raises the question whether this company (which has not a bad reputation) is exceptional or whether this awareness is too easily taken for granted.

What stands out in this company as the most prominent reason not to adopt traceability, apart from awareness, is the organizational separation of development projects from the later phases of the software life cycle. Project managers are motivated by and rewarded for achieving the goal for which they are held accountable, which is delivering the right project on time and within budget. There is little incentive to use traceability when most of the benefits are outside the project.

Tender projects in which the supplier is to quote a fixed price are less likely to adopt traceability from a financial perspective – unless the client explicitly requires it.

The technical means for traceability exist. In our case study, the two big obstacles are a lack of awareness and the way software development projects contracted and organized. The latter is more difficult to change than the former. The expected implementation of Sarbanes-Oxley is a chance to change something for the better and to get traceability accepted at a larger scale.

This study adopted an explorative research approach. The findings resulting from one case study should be considered as our contribution to the understanding of decision making on requirements traceability by means of the conceptual articulations and practical insights for the subject matter. To enhance this basic understanding, follow-up research is suggested to test the proposed categories, factors and their relationships in other organizations. We recommend to including different roles, such as software architect, in follow-up studies. Too little is known about their influence on the process. Another interesting topic might be the influence of a development method, but then in the sense of it already proposing the use of traceability, therefore contributing to development organisation awareness. Once the foundation of decision making for requirements traceability is established, one can suggest the appropriate approaches, tools, techniques for adopting and implementing requirements traceability.

## References

1. Robertson, J. and S. Robertson (1999). *Mastering the Requirements Process*. Addison-Wesley, New York, N.Y.
2. Gotel, O. and A. Finkelstein (1994). An Analysis of the Requirements Traceability Problem. *Int. Conf. on Requirements Engineering (ICRE'94)* 94–101.
3. Arkley, P. and S. Riddle (2005). Overcoming the Traceability Benefit Problem. *13th IEEE Int. Conf. on Requirements Engineering (RE'05)* 385–389.
4. Ramesh, B. (1998). Factors influencing Requirements Traceability Practice. *Communications of the ACM* 41(12) 37–44.
5. Ramesh, B. and M. Jarke (2001). Towards Reference Models for Requirements Traceability. *IEEE Transactions on Software Engineering*, 27 (1) 58–93.
6. Hull, E., et al. (2005). *Requirements Engineering*. Springer Verlag, London, Second Edition.
7. INCOSE. (2006). *INCOSE Requirements Management Tools Survey*. International Council on Software Engineering, Retrieved March 6<sup>th</sup> 2006, from <http://www.paper-review.com/tools/rms/read.php>.
8. Lindvall, M. and K. Sandahl (1996). Practical Implications of Traceability. *Software – Practice and Experience* 26 (10)1161–1180.
9. Stout, G. A. (2001). *Requirements Traceability and the Effect on the Systems Development Lifecycle*. Revere Group whitepaper.
10. Dorfman, M. and R. Chardon (2001). Early Experience with Requirements Traceability in an Industrial Environment. Industrial presentation, 5th IEEE International Symposium on Requirements Engineering (ISRE'01)
11. Daft, R. L. (2000). *Management*. Dryden Press, Fort Worth, TX, 5<sup>th</sup> Edition.
12. Matheson, J. E. and R. A. Howard (1983). An Introduction to Decision Analysis. In: R. A. Howard and J. E. Matheson (Eds.), *Readings on the principles and applications of decision analysis I.. Strategic Decisions Group, Menlo Park, CA, 17–55*.
13. Aydin, M. N. (2006). *Decision-Making and Support for Method Adaptation*, Ph.D. Thesis, University of Twente, Enschede, the Netherlands, 2006.
14. Howard, R. A. (1983). The Evolution of Decision Analysis. In: R. A. Howard and J. E. Matheson (Eds.), *Readings on the principles and applications of decision analysis. . Strategic Decisions Group, Menlo Park, CA, 5–16*.
15. Rational (2006). *Rational Unified Process*, version 2003.06.15, IBM.
16. OGC (2005). *Managing Successful Projects with PRINCE2*. The Stationary Office, London, 4th Edition.
17. ISO 9000-1 (1994). *Quality systems – Model for Quality Assurance in Design, Development, Production, Installation and Servicing*. International Organization for Standardization
18. Carnegie Mellon SEI (1999). *The Capability Maturity Model, Guidelines for Improving the Software Process*. Addison Wesley, Reading, Massachusetts.
19. US Congress (2002). *Sarbanes-Oxley Act of 2002*. Washington, USA, Congress of the United States of America.
20. Antonioli, G. (2002). Recovering Traceability Links between Code and Documentation. *IEEE Transactions on Software Engineering* 28 (10) 970–983.
21. Jarke, M. (1998). Requirements Traceability. *Comm. of the ACM* 41 (12) 32–36.
22. Yin, R. K. (1994). *Case Study Research; Design and Methods*. Sage Publications, Thousand Oaks, 2nd Edition.
23. Lauesen, S. (2002). *Software Requirements: Styles and Technique*. Pearson Education Ltd.