The Use of Workshops for Requirements Engineering

Master Thesis

A.B. Belgraver

Master Software Engineering

Universiteit van Amsterdam

Supervisor: Hans Dekkers

Approval date: 2007-09-03

Contents

Abstract

On average only 26% of projects are completed on time, within budget, with all features as specified. For larger management information system development projects over a third will be cancelled before completion. It is estimated over half of all system defects are introduced during requirement engineering. Clearly there is a problem.

Literature suggests getting executives, users and other stakeholders together in a guided session (a workshop) to collaboratively set goals and determine requirements is an effective way to quickly create high quality requirements. Prior research often found that working in groups is difficult and does not improve the quality of the product.

To find evidence of process improvement we tried collecting empirical data. We found organisations were hesitant to participate, due to the private nature of required information and not having the information we needed. We could not gather the required data.

We did find that workshops are often used for the creation and modification of management information systems. Larger IT organisations used workshops as default technique. They are more often used for creating vision, setting business goals, and scoping, than actual specification of requirements. Workshops are attended by four to twelve participants, last two to four hours, and are held two to three times a week.

Based on literature study, interviews, and a survey, we conclude that workshops are a valuable tool for requirement elicitation. However, getting the right people to participate is difficult. Also skillfull facilitation is required to prevent problems and unproductive sessions. To succeed, workshops demand extra skills from users and designers. Nevertheless, attaining benefits is certainly feasible, and workshops are seen as being very suitable for finding requirements by practitioners.

The magnitude of current day projects means significant benefits can be found in using cross-functional teams and having users participate in system design. Workshops provide a way to execute group elicitation tasks, offering the possibility to save effort, and increase requirement quality reducing costly rework activities.

Preface

This thesis is the result of the master research project carried out at the University of Amsterdam during the months of April - July of 2007. Its objective was to research the use and productiveness of workshops for requirement engineering.

Based on prior experience and recent coursework I realised that much, if not most, can be won or lost during the precious early stages of a software development project. No amount of skillful programming and testing can save a project when its goals and objectives are not correct. Fragile ideas can be made or lost easily during the Inception phase of a project.

As one interviewee said: *"The real requirements will surface sooner or later anyway, so you better make sure you find them sooner rather than later."*

The results of this master research project would not have been possible without the help of many people. I would like to take this opportunity to thank my supervisor Hans Dekkers for his valuable and insightful feedback. I also thank Dick Croes, Huib van der Meijden, Mark Hoogenboom, Mats Fillerup, Mike van Spall, Karel Sommer, and everybody who helped, for their valuable time and information during the interviews and participation in the survey.

During the Master Software Engineering many teachers provided valuable insights into the world of Software Engineering. I am grateful to Paul Klint, Jan van Eijck, Jurgen Vinju, Peter van Lith, Hans van Vliet, and all guest lecturers.

It is my sincerest hope that you, the reader, will find this thesis as fascinating as I have found discovering the many facts and aspects of requirement engineering and workshops over past few months.

Alexander Belgraver
Santpoort Noord
The Netherlands
August 14, 2007

Chapter 1

Introduction

On average only 26% of projects are completed on time, within budget, with all features and functions as originally specified. (Standish Group, 2000) For larger management information system projects, between 39% (10.000fp) and 48% (100.000fp) will be cancelled before completion. (Jones, 2000) In 1998 these failed projects cost U.S. companies an estimated $75 billion dollar, and an estimated $22 billion dollar in cost overruns. Challenged and failed projects are the norm. (Standish Group, 2000)

These days people spend more time with a computer than with their significant other, while describing their most recent experience with a computer problem as one of anger, sadness, or alienation. (Kelton Research, 2007) Cancelled and failing projects cost billions of dollars, cause frustration, burn-outs, and even death. (Standish Group, 2000; Yourdon, 2004; Leveson and Turner, 1993) Failing projects have a big and ever increasing impact on our lives.

Requirements errors and deficient requirements are the biggest single cause of problems and failure. It is estimated that between 40% and 60% (Leffingwell, 1997; Wiegers, 2001; EBG Consulting, 2007) of all defects are introduced during requirement engineering. Finding and fixing these requirements defects accounts for 70 to 85% of the total rework cost. Fixing these defects during acceptance tests has a 50:1 cost increase over correcting them during the requirements phase. Fixing requirement defects as maintenance activity has a 200:1 cost increase. (Leffingwell, 1997) Defect removal can costs 53% of total project expenses. (Jones, 2000) Getting the requirements right early has huge benefits for the well-being of the project, its personnel and its users.

The top three project success factors are user involvement, executive management support and clear business objectives. (Standish Group, 2000) Getting users, executives and other relevant stakeholders together in a guided

workshop to analyse problems, set objectives, and define and design the solution are best practice way to quickly create stable and high quality requirements. (Wood and Silver, 1995; August, 1991; Gottesdiener, 2002; Hoogenboom et al., 2004; McConnell, 1996)

A stakeholder is "a person, group or organisation that is actively involved in a project, is affected by its outcome, or can influences its outcomes." (Wiegers, 2003) These often include users, executives and developers.

The adoption rate of workshops in software development has grown. Jones (2000) indicates that 70% of inspected projects in excess of 100 function points (a measure of software size) used workshops for gathering and analyzing requirements.

While the adoption of workshop techniques increases, surprisingly little research, benchmarks or assessments about requirement workshop effectiveness could be found. Also, descriptions on how workshops should be used, and how workshops are used in practice, also vary between publications and practitioners. Literature (e.g. McConnell (1996); Wiegers (2003); Glass (2003); Robertson and Robertson (1999); Gottesdiener (2002)) classifies requirement workshops as best practice, but besides anecdotal evidence almost no empirical evidence is provided to support its use.

One study found that only 15% of investigated projects using workshops had user satisfaction and buy-in to the system as specified, only 10% of projects indicated that requirements were defined faster, and only 5% reported on achieving consensus on requirements. (Davidson, 1999) Another study also found that users did not experience significant improvement on consensus management and acceptance compared to traditional design methodologies, such as using normal interviews. (Purvis and Sambamurthy, 1997)

Also research in psychology found that group work is often less productive. Various problems are associated with group work, such as motivation and coordination problems, biased information sharing, and increased vulnerability to cognitive biases and errors. (Nijstad and Stroebe, 2006) Guiding a group to a shared solution includes overcoming cognitive, social and linguistic problems (Nuseibeh and Easterbrook, 2000) while trying to lead the process forward in problem solving, solution finding and requirement definition.

Unlike common expectations, research found that idea generation in groups result in fewer ideas and fewer good ideas than individual activities combined. (Diehl and Stroebe, 1987) In fact, productivity loss is observed consistently in brainstorming groups with more than two members. (see Sutton and Hargadon (1996) for references)

Bringing stakeholders with conflicting demands together in important projects can also easily mean disaster. When workshops do go wrong, it can have

devastating effects on the project and on the organisation. It can cause hostility between participants that persists for years after the workshop ends. (Davidson, 1999)

The central research question is therefore stated as:

"Are workshops a valuable tool for requirement elicitation, or an added project risk?"

This document describes the research project to answer the above question. It is divided in four parts.

Part 1: Problem and Context
The problem and context is explored in this chapter and in Chapter 2.

Part 2: Research Strategy
Chapter 3 explains the research strategy.

Part 3: Research Execution
In Chapter 4 we discuss the literature study. Chapter 5 holds a summary on our interview findings. Chapter 6 presents the results of the requirement workshop survey. Chapter 7 is used to discuss the difficulties in collecting empirical data about requirement practices.

Part 4: Answers and Conclusions
Chapter 8 presents an overview of what requirement workshops are and how they are used. In Chapter 9 we discuss whether or not workshop benefits can be attained.

Finally, appendices include the annotated bibliography, more interview results, the questions that were used in the survey, an overview of the empirical data we tried collecting with an example from a project we analysed, and a discussion about the experiment of the shortening of the inception phase of a project.

Chapter 2

The Requirement Engineering Context

A software development project often starts with an idea for improving a situation. Defining the correct purpose for a software system to fit this idea, is difficult and prone to errors. Requirement gathering is the first engineering chance to fail in a software development project. (Firesmith, 2003) To illustrate, the software development lifecycle is shown in Figure 2.1.

A definition of requirement engineering is given by Zave (1997):

> Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families.

This definition tells us that requirement engineering is more than just taking down what some users would like to see in a system. It tells us that the goals of the people involved lead to the actual requirements. Engineers develop software not as an end-goal, but to achieve real-world goals.

Figure 2.1: The Linear Software Development Lifecycle

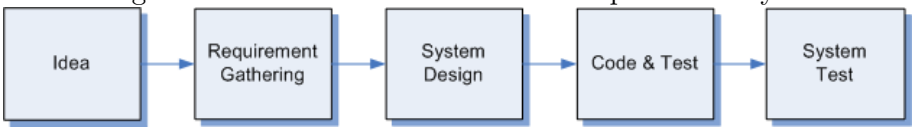
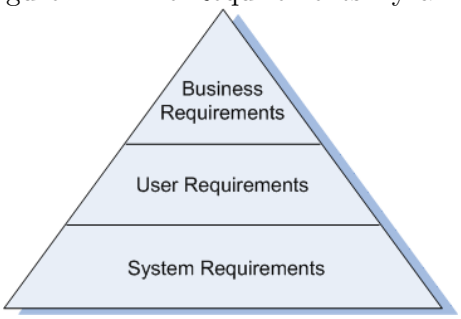


Figure 2.2: The Requirements Pyramid



The IEEE Standard Glossary of Software Engineering Terminology defines a requirement from a software engineering perspective as:

1. A condition or capability needed by a user to solve a problem or achieve an objective.

2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document.

3. A documented representation of a condition or capability as in 1 or 2.

The definition from the IEEE shows there is a difference between users needs and software capabilities. For users, business executives and system developers, requirements have a different meaning. Figure 2.2 shows the three types or requirements.

At the business requirements level, requirements engineers focus on why a system is necessary. They look at high-level objectives of the organisation and the customer, problem solving, goal setting, creating solution ideas, and setting scope. This includes finding or defining business rules, such as corporate polices, government regulations and industry standards. (Wiegers, 2003)

At the user requirements level, requirement engineers focus on users' goals and what is needed. It finds or defines goals, scenarios, and tasks in which the system is to be used, to describes what the users will needs to do to reach their goals.

Finally system requirements specify how the needs will be fulfilled. It details the functionality of the system that the developers should implement, and the quality demands on how developers implement them into the system. (Wiegers, 2003)

2.1 Activities

Many activities are performed to define requirements on these three levels. Requirement engineering consists of the following core activities: (Nuseibeh and Easterbrook, 2000; Wiegers, 2003; Davis and Hickey, 2002; Abran and Moore, 2004)

- Eliciting Requirements and Analysing Problems
  - Determine what needs exist by identifying problems, stakeholders and goals.
- Modelling and Analysing Requirements
  - Create models to aid in analysing the domain.
  - Determine which needs are to be addressed.
  - Select an appropriate solution from a variety of possibilities.
- Specification and Communicating Requirements
  - Document the intended external behaviour of the system.
  - Communicate the requirements to the stakeholders.
- Agreeing and Validating Requirements
  - Get agreement and commitment from stakeholders.
  - Validate that the solution solves the problem.
- Managing and Evolving Requirements
  - Steer the development process to the correct solution.
  - Manage the ongoing evolution of the user's needs.

2.2 Elicitation Methods

Gathering the actual requirements for a new development project is not easy. More than half of all software projects were compromised from the start by a failing requirement engineering process. (Standish Group, 2000) It is estimated that between 40% and 60% (Leffingwell, 1997; Wiegers, 2001; EBG Consulting, 2007) of all defects are introduced during requirement engineering. Finding and fixing these requirements defects accounts for 70 to 85% of the total rework cost. This demonstrates the failure of requirement engineering. (Davis and Hickey, 2002)

Thomas and Hunt (1999) stated the following about requirement gathering:

> The word 'gathering' seems to imply a tribe of happy analysts, foraging for nuggets of wisdom that are lying on the ground all around them while the Pastoral Symphony plays gently in the background. 'Gathering' implies that the requirements are already there-you need merely find them, place them in your basket, and be merrily on your way. It doesn't quite work that way. Requirements rarely lie on the surface. Normally, they're buried deep beneath layers of assumptions, misconceptions, and politics.

Christel and Kang (1992) define three categories of requirement elicitation problems:

- Problems of scope: Often the boundary of the system is ill-defined and requirements may address too little or too much information. "Avoiding contextual issues can lead to requirements which are ... unusable. Focusing on broader design activities improperly emphasizes developers issues over the users needs".
- Problems of understanding: Often cognitive problems are found. Problems with mutual understanding within groups as well as between groups such as users and developers, and omitting obvious information, that is often not obvious to other stakeholders.
- Problems of volatility: The changing nature of requirements, the developing organisation, and new insights during execution of the project cause project scope to change, and often grow, during the execution of a project.

Requirement engineers have a toolkit of techniques at their disposal, and need to know if, how and when these methods are effective. (Hickey and Davis, 2003; Davis and Hickey, 2002) Jantunen (2005) categorises these techniques in five categories:

- Traditional methods: introspections, interviews and questionnaires.
- Observational methods: observations, ethnographic studies, protocol analysis and contextual inquiry.
- Analytic methods: requirement reuse, documentation study and logging actual use.
- Prototype methods: prototyping, scenarios and storyboards.
- Group elicitation methods: focus groups, brainstorming and workshops.

2.3 Group Elicitation Methods

This research project focuses on group elicitation methods, and facilitated workshops in specific.

Focus groups are a form of qualitative research, a "moderated discussion among 8 to 12 users or potential users... A typical focus group lasts about two hours and covers a range of topics that you decide on beforehand." You can learn about "users' attitudes, beliefs, desires and users' reactions to ideas or to prototypes." (Usability.gov, 2007)

Brainstorming was originally created by Alex Osborn, and defined as "a method by which a group tries to find a solution for a specific problem by amassing a list of ideas spontaneously contributed by its members." (Hyde, 2005) As Osborn promoted, group size is between 6 and 12 participants.

"A facilitated workshop is a structured approach to ensure that a group of people can reach a predetermined objective in a compressed timeframe, supported by an impartial facilitator." (DSDM Consortium, 2003) Workshops often use low-tech visual aids such as flip-charts, brown paper, whiteboards, sticky notes and stickers.

In requirement workshops, "guided by a session leader, users and information systems professionals design systems together in structured group sessions. [It] harnesses the creativity and teamwork of group dynamics to define the users' view of the system - from the system objectives and scope through screen and report design."(August, 1991)

Requirement workshops are often held for brainstorming, focus groups, creating and evaluating prototypes, creating and testing scenarios, and performing other elicitation methods. Requirement workshops are held in the first three steps of the software development lifecycle as shown in Figure 2.1. This includes developing the original idea further, creating the requirements and designing the system.

Literature suggests that workshops increase involvement of stakeholders, increase the quality of requirements, and reduce the total duration of the project. (August, 1991; Wood and Silver, 1995) It could halve implementation effort, and save hundreds of thousands of dollars per project. (Carmel et al., 1993)

Research however found group processes are often less productive and produce less quality than working individually. (Diehl and Stroebe, 1987) In fact, productivity loss is observed consistently in brainstorming groups with more than two members. (see Sutton and Hargadon (1996) for references) One research project found that only 15% of investigated projects using workshops had user satisfaction and buy-in to the system as specified. (Davidson, 1999)

Chapter 3

Research Method

The main research question for this research project was stated as:

"Are workshops a valuable tool for requirement elicitation, or an added project risk?"

3.1 Research Questions

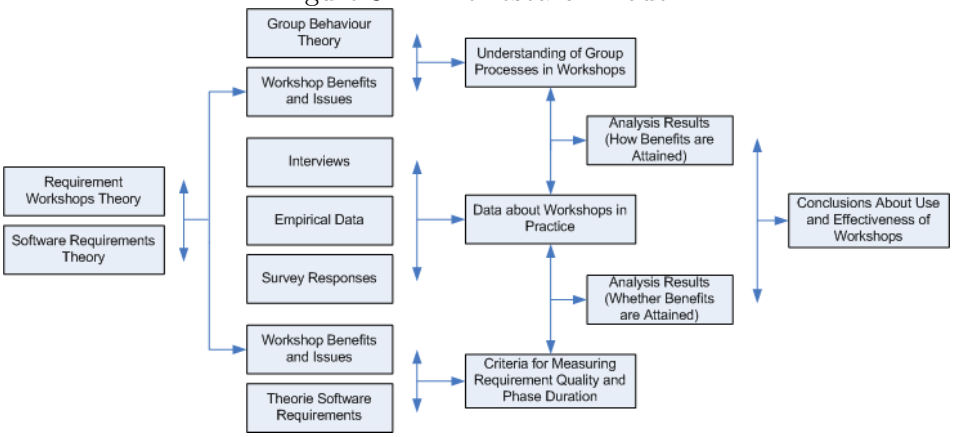
To answer the main research question, five derived questions were created. Based on literature study, we found different definitions and methods for using workshops in requirement development. To create a clear answer as to what requirement workshops really are, we stated two derived questions:

1. What are requirement workshops?
2. How are requirement workshops used?

Many benefits for workshops are presented in literature. We categorised these benefits, as presented in Appendix A.4 on page 85. Based on these categories we stated questions to see if these benefits are actually found in practice.

3. Do workshops involve the user?
4. Do workshops increase requirement quality?
5. Do workshops increase productiveness of requirement development?

Figure 3.1: The research model.



3.2 Research Strategy

A research strategy was created to answer the research questions. Figure 3.1 show the strategy in a research model. (Verschuren and Doorewaard, 2000)

A study of requirement workshop theory and software requirement theory results in an understanding of the requirement engineering context and the workshop subject. Analysis of group behaviour research, combined with workshop theory, creates an understanding of group processes that are relevant to workshops.

Criteria for measuring whether the benefits of workshops are found in practice were based on a combination of workshops theory, software requirements theory and extensive brainstorming.

Based on literature study, interviews, empirical data, and the survey, information about workshop use in practice is obtained. This information is analysed to assess whether benefits are actually realised in practice. The data about practical use is also contrasted to the understanding of group processes to assess how workshop benefits could be realised in practice.

The combination of how benefits could be realised and whether they are found in practice, results in conclusions about workshop effectiveness for requirement elicitation and answers the research questions.

To execute the research plan and answer the questions, we used four main activities. Table 3.1 shows how these activities help to answer the questions. The following sections describes the activities.

Table 3.1: Question x Activity Matrix.

	Lit. Study	**Interviews**	**Survey**	**Empirical Data**
Q.1	x	x		
Q.2	x	x	x	
Q.3	x	x	x	
Q.4	x		x	x
Q.5	x		x	x

3.3 Literature Study

3.3.1 Goal

The goal of the literature study was to find out what literature says about the stated questions.

3.3.2 Method

To answer the research questions books and research papers were studied. A total of 17 books and 35 papers were studied, and more resources were checked for relevant information. Table 3.2 show in which categories books and papers were studied. Chapter 4 and the annotated bibliography in Appendix A list the most relevant resources in more detail.

As starting point for literature study, the current day reference work 'Requirements by Collaboration' by Gottesdiener (2002) and 'Workshops' by Hoogenboom et al. (2004) were used, as advised by early contact with experts.

To make sure that all important and relevant resources were identified and studied, the following techniques where used:

- The bibliography and listings of further or related reading were studied of relevant publications.
- Four leading on-line databases and the Internet were searched for relevant publications.
- The survey was used to ask participating facilitators about relevant resources.

Table 3.2: Categorisation of literature.

Book Categories	**Number Studied**
JAD and Workshops	5
Requirement Engineering	3
Software Estimation, Measurement and Assessment	3
Software Engineering (other)	6
Research and Scientific Writing	7
Paper Categories	**Number Studied**
JAD and Workshops	5
Facilitation and Collaboration	8
Group Processes	11
Requirements Engineering (other)	11

3.4 Interviews

3.4.1 Goal

The goal of the interviews was to help answer the stated questions by exploring the use of workshops in requirement processes in practice.

3.4.2 Method

To reach this goal, a total of 22 interviews were held with a wide variety of professional facilitators and requirement engineers. Interviews also included an expert on collaboration engineering, a representative of a requirement management tool, a business administration professor, and a professional on project planning and estimation.

Early interviews were of an open and exploratory nature, focusing on the organisation's requirement processes, and what interviewees found important. Later interviews were more directive, and were also used to validate hypotheses and theories found in earlier interviews. Structurally, the interviews were based on the structure of the survey to assure all relevant subjects were discussed.

The duration of these interviews varied from one hour to two hours. The interviews all took place in the Netherlands. With the exception of one interview, all interviews took place in person, face to face. Prior to the interview, it was established that the organisation and the interviewee used workshops in the requirements phase.

3.4.3 Participants

As starting point for finding respondents, a list of participants of a requirements knowledge sharing session of the DSDM organisation was used. Based on this list, a first set of organisations and contacts was found. Contact was made over telephone with 31 organisations, only counting those that indicated using requirement workshops, to the find the right interview candidates. A further set of interview candidates was found in conversation with representatives of contacted organisations, and by asking interviewees. As motivating factor to participate in interviews, candidates were promissed to receive the research results.

In total 15 interviews were held with requirement facilitators or managers of requirement facilitators in 12 different organisations, as shown in Table 3.3.

Table 3.3: Interviewees and organisations

Type	**Organisations**	**Interviewees**
Requirement Specialists	2	2
Internal IT departments	4	5
Small IT organisations	2	4
Large IT organisations	4	4

3.5 Survey

3.5.1 Goal

The goal of the survey was to help answer the questions:
Question 2: How are requirement workshops used?
Question 3: Do workshops involve the user?
Question 4: Do workshops increase requirement quality?
Question 5: Do workshops increase productiveness of requirement development?

The survey was used to collect information about the use of workshops in practice, about experiences of facilitators with requirements workshops, and to find out what happens during these workshops.

3.5.2 Method

To collect data, we compiled a list of 53 questions in seven categories: workshop use, dimensions, preparation, session content, follow-up, quality and personal experience. The questions are shown in Appendix C, on page 93.

To make sure the right questions were asked:

- Questions were based on extensive literature study, and survey theory.
- Were based on brainstorm sessions with an experienced requirement engineer.
- A trial survey was created in which five workshop facilitators were invited to complete the survey and give feedback. Three responses with feedback were received, and feedback was incorporated.

3.5.3 Tools

The survey was held using a web based survey tool. The survey tool was chosen based on the following criteria: user friendliness of the survey, professional appearance, configurability of survey and survey attributes, useful reporting facilities, and a general impression of product stability. Two tools were selected and tested. The best was installed at a leading international hosting provider, and configured and personalised for use.

3.5.4 Respondents

As with the interviews, the starting point was the list of participants of a requirements knowledge sharing session. We contacted these organisations to find the right people in the organisation to participate in the survey, and screen them for suitability. More respondents were found by asking survey respondents and interviewees for further contacts. As motivating factor to participate in the survey, respondents were promissed to receive survey results if they completed the survey.

To make sure the right people were invited, the survey participants were screened for having experience in facilitating requirement workshops prior to receiving the survey invitation. This screening was often done over telephone, but also in person, or by recommendation.

A total of 66 facilitators were invited to participate, as shown in Table 3.4. Respondents, almost all of Dutch origin, came from a variety of different organisations, including several large and smaller automation organisations, requirement engineering organisations, organisations from the financial, insurance and health sector, and with research backgrounds.

Table 3.4: Survey respondents.

Category	**Number**
Invitations Sent	66
Survey responses	28
Participating organisations	17

3.6 Empirical Data

3.6.1 Goals

Empirical data was to be collected to find evidence for:
Question 4: Do workshops increase requirement quality?
Question 5: Do workshops increase productiveness of requirement development?

3.6.2 Method

To assess whether the benefits of workshops are realised in practice, we studied literature to find information about measuring requirements and requirement engineering. No applicable ways of collecting this data was found.

We created a way to measure the quality of requirements by measuring changes made to requirements. We measured productivity by measuring effort in preparation, execution and follow-up of workshops and the number of requirements that were created. The research model is presented in Chapter 7 on page 41.

Based on the created measurement mechanism, plans were divided into three sets of metrics: overall project data, requirement volatility data, and workshop data. The complete set of metrics is shown in Appendix D.

3.6.3 Requirement Metrics

To gather data, we contacted 31 organisations. Contacted organisations included both large and small software development organisations, requirement engineering specialists, in-house development organisations, and two requirement management manufacturers to try to get access to their customers.

Table 3.5 shows that only two organisations indicated they could deliver relevant information, and of three projects information was gathered. Con-

tacted organisations were hesitant to participate. This was often due to the sensitive information requested, or due to the fact that the requested information was not logged or registered. Only partial information could be found. We can conclude that we failed to collect the empirical data required to assess whether workshop benefits are found in practice. More information about collecting empirical data is presented in Chapter 7 on page 41.

Table 3.5: Metrics

	Number
Contacted Organisations	31
Participating Organisations	2
(Partial) Project Metrics Gathered	3

Chapter 4

Literature Study

To answer the research questions books and research papers were studied. A total of 17 books and 35 papers were studied, and more resources were checked for relevant information. This chapter presents and analysis of the most important resources, and whether research questions were answered by studied literature. Our findings about workshops are presented in Chapter 8, while more details about studied literature can be found in Appendix A.

4.1 Results

Two reference works for JAD can be found: 'Joint application development' by Wood and Silver (1995) and 'Joint application design' by August (1991). Both books talk about the same method, but August differentiates between two types of JAD workshops (JAD/Plan and JAD/Design) and talks about workshop customisation. Wood and Silver only discuss the JAD/Design workshops and its phases. As such, both books seem to cover slightly different methods.

The workshop books of 'Requirements by Elaboration' by Gottesdiener (2002) and 'Workshops' by Hoogenboom et al. (2004) are both more recently written. Where Hoogenboom et al. focus on practical aspects of facilitating workshops in software development, Gottesdiener focuses more on requirement workshops in specific.

Studied books about JAD and workshops used different names and models, and describe different processes. They could be seen as complementing each other but real contradictions have been found. The major difference is that both JAD books discuss longer sessions with different activities, whereas Gottesdiener and Hoogenboom focus on a series of activities and other activities. Also, changes are also introduced due to progress of time.

The studied books give an extensive amount of impression on JAD and requirement workshops. Books are full of information, but little analysis and objective evidence is presented about whether the benefits are attained. Also information practical application is more often about facilitation of workshops, than how they fit in software projects. Benefits and issues are described, though most books focus benefits. As evidence, these resources provide anecdotal reports. No research studies are presented about the effectiveness of workshops.

Other books have been checked for information about the use and effectiveness workshops. Including assessment and benchmarking references of Barry Boehm and Capers Jones. Surprisingly little benchmarks and assessments about requirements effectiveness and requirement workshops could be found. Jones (2000) did find that requirement volatility decreases when workshops are used, but he presents little information on which grounds his findings are based. Other books (e.g. McConnell (1996); Wiegers (2003); Glass (2003); Robertson and Robertson (1999)) do list JAD workshops as best practice and advise using them, but do not provide evidence that it actually works.

Of the studied research papers, again we find that few papers objectively measured the efficiency of workshops. Also, techniques of workshops used are not detailed. With all the different uses for workshops, little is know about what is actually measured.

Only one research project was found (Schalken et al., 2004) that measured projects to see the effectiveness. Two other papers based their findings on interviews and perceptions. Furthermore, some case studies of individual projects and workshops were found.

Based on our literature study, we conclude that while some research was done, little objective evidence is published about measuring requirements, the requirement phase and the use of workshops. No differentiation is made about which type of workshops are measured. Publications do state benefits, but besides giving logical sounding reasons no evidence is presented besides anecdotal information. The magnitude of (potential) problems are under presented in the books, though described more clearly in research papers.

4.2 Analysis and Validity

4.2.1 Analysis

We analyse whether our literature study answered our questions.

Question 1: What are requirement workshops?
Literature had a lot of information about workshops and requirement work-

shops. However, no clear answer was found about what exactly requirement workshops are. What workshops are used for, and how they are used in software projects, differs from publication to publication. We analyse contradictions between resources, and between theory and practice, in the survey results and interview results chapters.

Question 2: How are requirement workshops used?
Again, literature does not give a clear answer. Practical aspects as preparation, execution and follow-up are discussed, but significantly less about how workshop fit in the bigger picture of software development. Books on JAD also differ with other books about workshops on how they are used in projects.

Finally, *Questions 3-5* of whether benefits are realised. Some research has been done about JAD workshops, but they are largely about impressions and opinions. Only one research project measured objectively, focusing on whether productivity increases. We discuss answers to questions 3-5 in Chapter 9.

4.2.2 Validity

Was the right literature studied?
Based on early contact with facilitators we found relevant reference works. Related and relevant literature was found by:

- Studying the bibliography and listings of further or related reading were studied of relevant publications.
- Searching four leading on-line databases and the Internet for relevant publications.
- We asked interviewees and survey respondents about relevant resources.

All relevant literature that was found was categorised and checked for relevance to our topic.

To validate the completeness of our literature study, we:

- Compared our studied literature to literature referenced in relevant publications and studies.
- Asked experts reviewing our thesis if we missed important resources.

Based on these activities, we conclude our set of studied literature was validated.

Was our interpretation correct?
We internally validated findings by comparing between books. Furthermore, we externally validated the literature study with the survey and interviews.

Information about external validation is found in the survey and interview chapters.

We sent concept versions of relevant chapters to facilitators and other experts. We received feedback from seven experts with backgrounds in requirement workshops, requirement engineering, collaboration, and facilitation.

Feedback included some new insights and additional discussion based on reading the report. This information was added where applicable. Other feedback corrected our interpretation of studied literature regarding group psychology. Feedback most often included advice on the way we formulated ideas and results.

4.3 Recommendations for Improvement

Although some literature about group psychology has been studied, more literature about this subject could have been studied. They hold relevant information about group work and processes, and problems and benefits experienced by working in groups.

Chapter 5

Interview Results

To goals of the interview was to get information about the requirement processes in which workshops are used in practice. The first sections lists a summary interpretation of key findings, with a more detailed information in Chapter 8 and in Appendix B. The second section of this chapter holds analysis, validation and recommendations for improvement.

5.1 Results

Workshops are used for a wide variety of activities, and consist of many different types of building blocks. In practice, roughly two methods for using workshop activities were found: the Collect, Consolidate and Confirm method, and the Why-What-How method.

The Collect, Consolidate and Confirm method consists of three steps. First: Fact-finding workshops are used to collect relevant information from the stakeholder. These workshop are used as a source of information. Second: The consolidation activities are often performed without workshops, and are used to process and refine the collected information and create models, scenarios or prototype solutions. Finally: the third step consists of a confirmation workshop. These workshops are used for verification, validation and getting agreement by using presentations, perhaps showing protypes, and asking feedback.

The Why-What-How method follows the Requirement Pyramid model as shown in Figure 2.2 on page 6. Three different workshops can be found: The Why-Workshops are used to find the goals for the new system. What workshops are used to discover what is necessary to meet goals, and which user activities are required. Finally, How-Workshops are used to determine how the system supports the user in execution of his activities, and specify more detailed requirements.

We found that workshops were often held over multiple weeks, with between one and three workshops per week, lasting only a few hours. A series of workshops often has a fixed set of core participants. Project team (or project organisations) are a familiar way of changing organisations and key members, such as project leader, key consultants, key executives, and a requirement engineer tend to participate in these workshops by default. Additional experts and sources of information, such as customers, users, subject experts, and process experts are invited only as required.

We detected a split between business analysis (project inception) and user requirement analysis (specification and design). Requirement engineers and IT analysts are often called in after the solution idea has been created. When these projects reach IT departments, high-level requirements have already been 'pre-created'.

Often these high-level requirements are not of sufficient quality, creating the necessity to take a step back. This step back is often hard to sell to stakeholders and executives, but important for creating the right system. A symptom of this split is when one type of analysts and facilitator are used for the first workshops, and a completely different set in other workshops. This presumes that business analysis is done, which is not always the case.

Workshops are used differently for the creation of new systems than for the modification or replacement of existing systems. For the latter type of projects the scope and goals are often seen as more fixed. Though less workshop intensive, it is advised to still execute goals and scope workshops to prevent the almost inevitable discussions. In these projects workshops are more often used as verification and validation tools. Most interviewees indicated that workshops had more value for the creation of new systems systems, though as expected significantly less of these projects existed.

During interviews we learned about the Accelerated Solution approach. These workshops have been engineered to be a highly intensive, focused and facilitated series of collaborations in a special environment suited to collaboration. Weeks or even months of workshops are compressed into a few days.

Of all contacted organisations that used workshops, those with a well developed requirement process used workshops extensively, where organisations with a less well development processes used workshop techniques significantly less. These organisations often only used workshops techniques when the individual felt it was a good idea and happened to have knowledge and experience with them.

All interviewees indicated also using interviews and analysing existing documents during requirements analysis. Finally, workshops are a tool that is also applicable to problem solving and creating vision outside of requirements and IT projects.

5.2 Analysis and Validity

5.2.1 Analysis

Information retrieved from interviews were most often of a nature of personal experiences, and organisational processes. All individuals interviewed used different methods and processes. This means that contradictions in answers were most certainly seen. The descriptions present in this thesis are a synthesis of all information found.

Were the research questions answered:
Question 1: What are requirement workshops?
The interviews were used to validate our interpretation of literature with the experiences of workshop facilitators. Interesting viewpoints and insights about workshops were found, which are included in this thesis.

Most surprising, unlike our understanding of literature use cases and requirement statements were not often created during workshops. Often workshops were used to gather information, and in some situations define abstract descriptions, but scenarios, use cases and requirement statements were specified after the workshop. Workshops were also used to prioritise, verify and validate the created use cases and requirement statements. The survey supports this impression, as developing use cases were a less often used as activity, but were often used as model. Prioritisation was also often done in workshops.

Question 2: How are requirement workshops used?
Information was gathered about how workshops are used. Results are discussed above, and in Chapter 8. JAD theory suggests multiday workshops. These were not often found in practice. All but one interviewed organisations used workshops in a series of sessions lasting two to four hours. Two participants gave reasons, first: facilitator stamina, being sharp and alert for a whole day is exhausting, let alone consecutive days. Second reason: participants can more easily be invited for shorter sessions than longer sessions.

Theory suggests using outside facilitation as they are more objective. Interviewees indicated that a facilitator should have knowledge of the domain, context and subject. A facilitator that continually asks questions to familiarise themselves with the content, tends to be annoying to the participants and therefore counter productive. In some situations extra requirement analysts and domain matter experts are invited to offset deficient knowledge.

Question 3: Do workshops involve the user?
Interviews gave another view on participation and involvement of users. All interviewees that were asked indicated that getting the user to participates

was difficult. Getting the right participants is however crucial for the success of the workshop.

Participants usually do not have extra time available as their jobs already fills all available time. Especially during the summer months, when participants individually plan vacations it is difficult to plan workshops. Also shortened workweeks cause participants to work on different days.

With management support and correct priorities this problem can be solved. Interview respondents indicated: "we usually find a way." These statements are supported by the survey, respondents indicated that getting the right participants was difficult, but often 'not impossible.'

5.2.2 Validity

Were the right people interviewed?
To make sure the right people were interviewed, the candidates were screened for having experience in facilitating requirement workshops prior to the interview. This screening was often done over telephone. During the early stages of the interview further validation was done as interviewees were asked to describe their requirement activities.

Were a representative set of people interviewed?
The 15 interviews with facilitators did not cover all ways of using requirement workshops. Nor was any of these organisations investigated in depth, or was one specific way of using workshops investigated in depth. Results are based on open and exploratory interviews.

The results are not representative for the entire field of requirement workshops, but do give valuable information (opinions and personal views) about facts. Because of the wide variety of organisations that participated in the interviews, a broad spectrum of requirement processes using workshops was seen. Although not rich in detail or verification, this 'big picture' certainly has value.

Were the right questions asked?
Early interviews were of an open and exploratory nature, focusing on the organisation's requirement processes, and what interviewees found important.

Later interviews were more directive, and were also used to validate hypotheses and theories found in earlier interviews. Structurally, the interviews were based on the structure and questions of the survey to assure all relevant subjects were discussed.

The interviewer did not have extensive experience with interviewing, but did prepare by studying some literature about holding interviews. Interviewees tend to give more socially acceptable answers in interviews, therefore the

open and exploratory nature the the interviews are well supplemented with survey findings.

Was our interpretation correct?
In one occasion transcripts of the interview were sent to the interviewee for verification, as quite a lot of information was discussed. In some situation follow-up questions were asked to verify our interpretation. In most instances our interpretation was validated by testing hypothesis and views with other interviewees.

To verify and validate our findings and interpretation, relevant chapters of this research project have been sent to experts for feedback. We received feedback from seven experts with backgrounds in requirement workshops, requirement engineering, collaboration, and facilitation.

5.3 Recommendations for Improvement

This series of interviews had an exploratory nature. For future interviews we would advise to also create a set of specific questions to get more clear and shared answers on predefined topics. These answers would also be more easily internally validated. My interpretation of interviews could have been validated better by sending reports to interviewees more often.

Chapter 6

Survey Results

The goal of the requirement workshop survey was to collect data about the use of workshops in practice, to determine what workshops are used for and how they are used.

6.1 Results

In each section of the survey one or more statements were given, which the respondents ranked on a five point scale: from strongly agree to strongly disagree. Results are shown in Table 6.1 on a three point scale.

6.1.1 Workshop Use

We asked respondents for which type of projects and software they would use requirement workshops. Table 6.2 show the results. Workshops respondents indicated they would use workshops most for the creation or modification of management information systems.

We asked for which requirements activities the facilitator would use workshops, Figure 6.1 shows the results. Requirement techniques used besides workshop included interviews (57%), analysis of existing documents and other desk research activities (50%), and prototyping (18%).

The questionnaire asked respondents to rank a set of predetermined reasons for using workshops. Respondents had to choice between 9 options. For each response, the top ranked position earned nine points, then eight for the second, until 1 point for the last placed reason. Results are displayed in Figure 6.2.

We asked which downsides respondents experienced. The most often reported problem was difficulty to organise workshops, particularly getting

Table 6.1: Questionnaire statement results.

Statement	**Agree**	**Undecided**	**Disagree**
Workshops reduce the requirement effort	48%	36%	16%
Interviews are better in finding and understanding new requirements	18%	39%	43%
Getting the right participants is often impossible	46%	15%	39%
Participants cannot judge correctness of information systems models	28%	20%	52%
It regularly happens real commitment is missing	36%	11%	53%
It regularly happens good facilitators get support and commitment for the *wrong* requirements	28%	28%	44%
Putting stakeholders with conflicting needs together will cause problems that increase project risks	16%	8%	76%

Table 6.2: Types of projects for which respondents would use workshops.

Type of Project	
Creation new systems	81%
Modification of existing systems	81%
Software product evaluation/selection	42%
Software product configuration	12%

Type of Software	
Management Information Systems	92%
Embedded Software	31%
User Tools Software	31%
Developer Tool Software	12%
Utility Software	12%
Systems Software	4%

Figure 6.1: Requirement activities done in workshops.

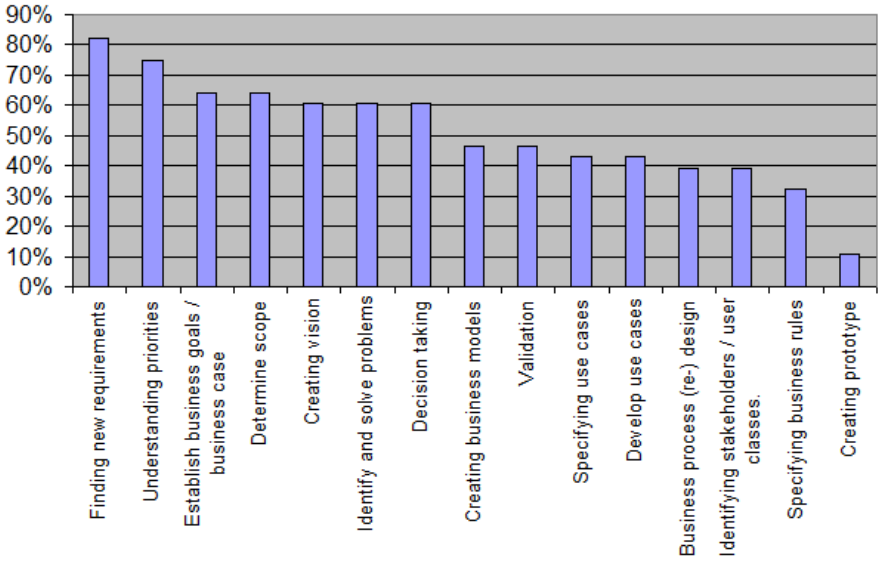


Figure 6.2: Reasons for using workshop [points scored]

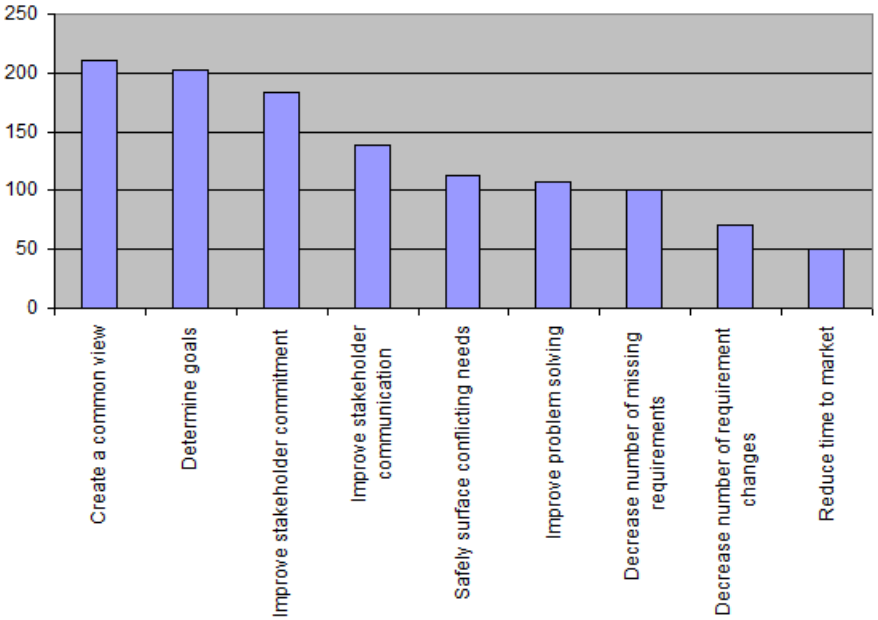


Table 6.3: Workshop Dimensions.

Workshop duration	2-4 hours
Number of participants	4-12
Number of requirement workshops in a project	2-8
Requirement phase duration	15%-25% of project duration
Time spent on workshops (incl preparation & follow-up) vs other requirement tasks	40%
Preparation effort for 8 hour workshop	15 hours
Follow-up effort for 8 hour workshop	10 hours

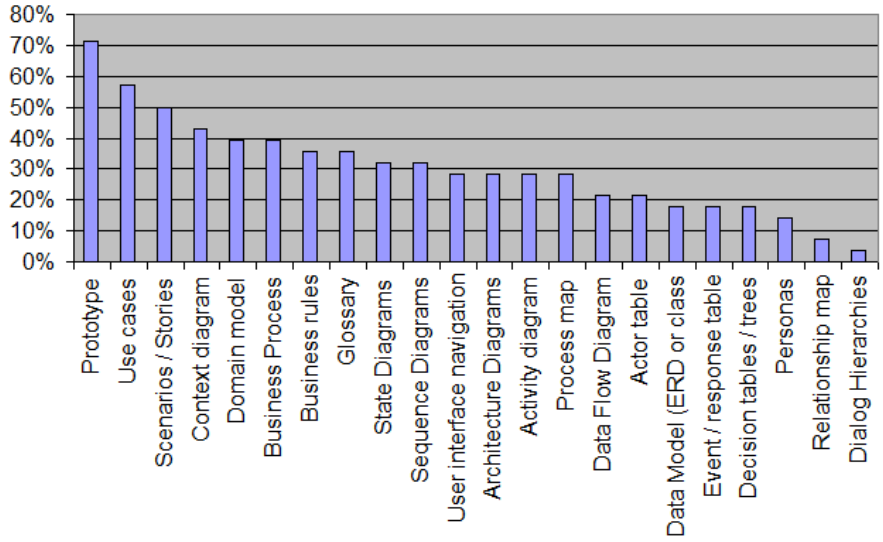
everyone to attend and be prepared. Another reasons cited multiple times was that they take a lot of time to plan. Less commonly reported downsides included that the facilitator is required to have an understanding of the context to be productive, that workshops are difficult to incorporate in fixed price/date projects, problems with mandate, difficulty to structure the outcome, extensive discussions, and that after capturing workshops results they must be analysed and improved.

6.1.2 Preparation

Facilitator preparation tasks included: holding interviews; investigate to understand context; desk research about the subject, background, scope and current requirements; get an understanding of the political situation (who is in charge, who might be eliminated by the system); stakeholder analysis, defining goals for the workshop; identify deliverables; design optimum workshop process and make agenda; prepare workshop supplies, templates, handouts and/or a presentation of known information; prepare the workshop room; and prepare participants.

In some situations participants of workshops have preparation work as well. These include clearing their schedule; reviewing relevant existing documentation, consulting their own organisation to have enough mandate to make decisions, analysing their own tasks and interview colleagues to draft requirements, and templates used in workshop. The quality of pre-work of participants was judged as weak (45%) or adequate. (42%)

Figure 6.3: Models used during requirement workshops.



6.1.3 Content

Respondents were asked which models they used during workshops. Results are shown in Figure 6.3.

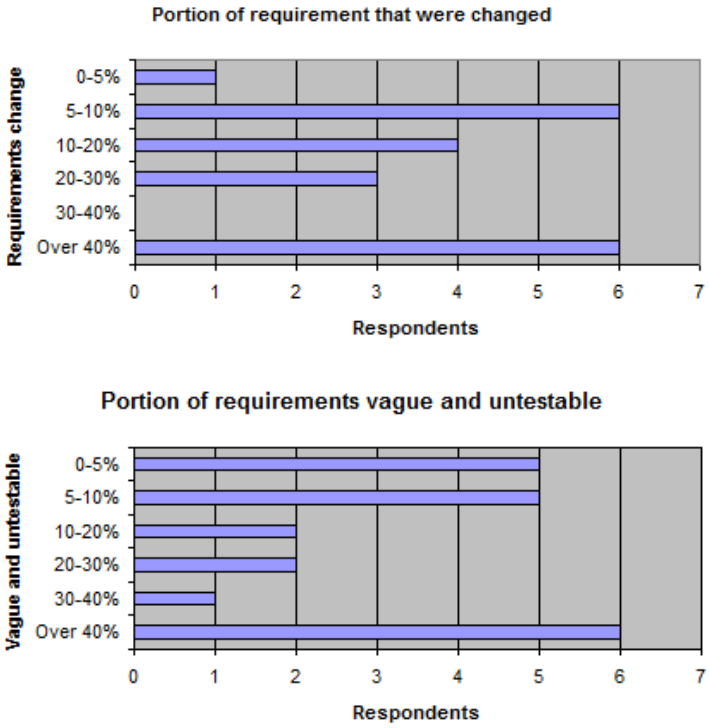
Other non-listed models that where mentioned included mind maps, rationale model, rich picture, process pictures (mixture of process diagram and IT architecture diagram), SWOT, balanced scorecards, organisation chart, free format diagrams, and several own developed methods, principles x requirements matrix, requirements x components matrix, planguage models, stakeholder onion model, benefits logic model (MSP), capabilities overview (MSP), product breakdown structure (Prince2). One respondent indicated that models should stay in the language of the domain, and not use UML.

We asked respondents what portion of the requirement generated in workshops was vague and untestable, and what portion was changed. Results are shown in Figure 6.4.

6.1.4 Follow-up

Respondents indicated that workshop deliverables are: depending on the subject of the workshop a list of issues, goals, scenarios, risks, requirements, use cases, business rules, actions, decisions, activities to do, different types of models, and so on; prioritised requirement lists; brown paper (or flip-charts) stickered with sticky-notes; information in the form of key words and one-liners; consensus; a common feeling and/or understanding; mutual

Figure 6.4: The quality of requirements.



agreement; and a list of participant expectations of the meeting (to be created at the start of the meeting that was held).

Respondents indicated follow-up activities for facilitators included transferring results (hard and soft) of the workshop to some form of document; making models; do traceability; structure information; detail use cases; send report (or other work products) to users for review and feedback; make sure activities are followed up on; guide, support and advise users; and evaluate the workshop. Planning a new session was also seen as a follow-up activity.

For workshop participants, follow-up consisted of attending to open issues and actions that were assigned during the workshop; reviewing the workshop report; giving comments and/or agreeing to the created content; get commitment from colleagues; and read and verify requirements.

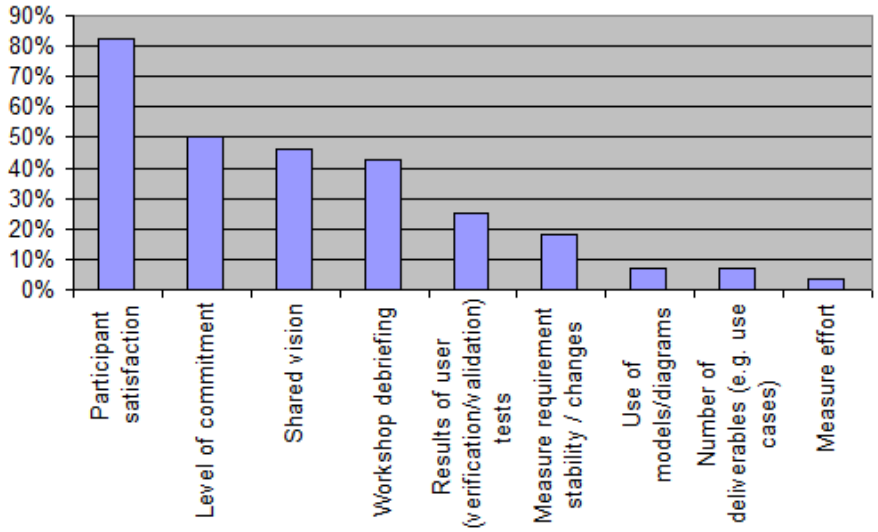
6.1.5 Quality

As an indication of when workshops are not to be best method, the most given answer was: when stakeholders cannot find the time to meet, when participants do not have mandate, or when a single (or very few) expert can state requirements. Other answers were: when reverse engineering an existing system; for extremely complex systems or batch systems; when participants have no authority or mandate; when the new IT system is chose on political grounds; high pressure projects; when participants have very different goals; and when important stakeholders are not accustomed to voice requirements.

If workshops could not be used, how would this impact quality. Answers were: stakeholders missing shared vision; commitment and common understanding would be less, scope could be less correct; and same quality, it would just take more time. One respondent indicated that the resulting solution would prove to be inappropriate after completion. Another respondent indicated that the quality of writing, and therefore understandability, would be better, but stakeholder commitment would be worse.

The evaluation of workshops were most often done by measuring participant satisfaction. Figure 6.5 shows the results of how workshops are evaluated by respondents. Other evaluation activities included: measuring the ability to move forward to the next steps; peer review of requirements; doing a benefits and concerns with participants; checking the quality of deliverables; finding satisfaction of executive sponsor; whether all stakeholders participated and show commitment; whether requirements lead to a project and whether that project succeeded.

Figure 6.5: How workshops are evaluated.



6.1.6 Personal

Respondents indicated that learned lessons included: resolve conflicting goals first; more specialised modelling; more care with facilitation; willingness to listen to 'irrelevant' details (it may just be of vital concern); to select the requirement tool that fits the need, workshops are just one tool; better selection of participants; focus more on business case than on technology; cancel workshops when authorised participants are not coming; accepting that there is not one correct solution, more adequate solutions exist; confrontation of stakeholder disagreements; listen to people at the 'coffee machine'; create matrices of principles x requirements to see if all principles are met and all requirements are required (introduce consistency checks); and not to stop until requirements are SMART. (Specific, Measurable, Achievable/Acceptable, Relevant/Realistic, Time-bound)

We also asked what training facilitators have had. Half of the respondents indicated having received formal facilitation training. Several had on-the-job training, but 29% indicated not having received training.

Respondents also evaluated the suitability of workshops for finding requirements. Results are showing in Table 6.4.

Reported books and reference material included 'Workshops' by Hoogenboom et al. (2004); Requirements by Collaboration by Gottesdiener (2002); Collaboration Explained by Jean Tabaka; Begin bij het eind met SMART requirements, Dijkgraaf and Van Spall (2007); Mastering the requirements process, Robertson and Robertson; Managing Requirements For Project

Table 6.4: Suitability of workshops for finding requirements.

Suitability	
Excellent	15%
Very Good	65%
Adequate	15%
Weak	5%

Success; Robertson and Robertson; books by Tom Gilb, 'Beweging in je brein', 'Het team als probleemoplosser', 'Doelgericht vernieuwen', 'Ideeen voor creativiteit', training material, and internet.

Respondents indicated they would like to know about: best practices; new ideas or views on workshops; more techniques and workshop instruments that can be used; the required level of detail of work during workshops; which techniques are used; best practices for requirement workshops; fit-for-purpose assessments; new techniques or methods; templates; and in which situation which kind of workshop should be used.

6.2 Analysis and Validity

6.2.1 Analysis

While analysing results, we were surprised at the level of workshop evaluation. Respondents did indicate evaluating satisfaction, but only half of participants measured commitment and a shared vision. Furthermore, little was measured about effort and productivity, which models were used, and whether that worked.

Several questions showed a surprisingly wide bandwidth and fairly even distribution in answers. These are: requirement activities in workshops, models used in workshops, number of participants, and number of requirement workshops in a project. We interpret this as evidence that workshops are used for a wide variety of tasks and activities. Unlike our expectations, and leading activity of 'finding new requirements' suggests, workshops were not often used for actually stating requirement and defining use cases. Also, 'finding new requirements' was rather all-inclusive, as a rather large set of activities can be seen as finding new requirements.

Based on interviews and literature, we would have expected more effort for preparation and follow-up. Also preparation and follow-up effort gave a wide variety of answers. For preparation answers varied from 0 to 80 hours,

standard deviation of 18. Follow-up varied from 0 to 120 hours, standard deviation of 35. We conclude three things:

1. Participants all perform different activities during the preparation and follow-up.
2. Even though we asked about effort after we asked about all tasks facilitators do in preparation, we presume our question of 'How many hours of preparation work would be required for a 1 day (8 hour) workshop?' was ambiguous as to what was included here and what was not.
3. The amount of work depends on more than just the duration of the workshop, but also the number of participants, the intensity and complexity of work during the workshops, and other factors.

The number of workshops was relatively low compared to expectations. Results indicate 2-8, again with a high standard deviation. If workshops are used to go from brainstorming and solution creation to iterating over prototypes, we would have expected a higher number of workshop. We hypothesize that workshops are used more often in early brainstorming and goal setting phases, and less often in later design phases. This hypothesis is supported by seeing more prioritising, problem solving, scope and vision activities than use case and prototyping activities.

When we compare our results with Davidson (1999), we find significantly less 3-5 consecutive day workshops. As this aspect of JAD is difficult to realise in practice, it has less often been adopted by other methods using workshops, such as DSDM. We hypothesize that this is an indication of the declined use of the JAD method. As Davidson, we also found fewer projects defined requirements in workshops, and workshops were used more often for high-level requirement activities.

Davidson found that often IT specific analytic models are used. Analysis shows 70% of our respondents indicated using at least one IT specific model. (70% indicated using at least one model of: ERD, DFD, state diagram, sequence diagram, architecture diagram, and activity diagram.) This seems to support findings.

Did the survey answer the research questions?

Question 2: How are requirement workshops used?
Yes, the survey does give an indication for which types of projects workshops are used, and a general idea about the dimensions of workshops. It presents a list of reasons for using workshops, a list of activities executed during workshops, and an overview of models that are used. We also found results about preparation and follow-up activities.

One apparent internal conflict in the survey results is the use of prototypes. Respondents indicated creating prototypes as activity least often, but most often as a model used. We presume that respondents did use workshops to validate prototypes, but not to create and change them. Perhaps we should have added the option 'validating prototypes' as activity for more information.

We hypothesize that while it is possible to create and change prototypes in workshops real-time, this is not often done. Prototypes are used most often for walk-throughs, and for obtaining feedback. Any modifications are done as follow-up work.

Question 3: Do workshops involve the user?
Whether or not users are feeling involved was not measured. This survey does give an indication as to how the user participates in system design during workshops. We found activities in which the user participates and models users help to create and verify. The survey also gives an indication on preparation and follow-up work the user often does when attending workshops.

Question 4: Do workshops increase requirement quality?
Answers about requirement quality gave very erratic results. A small set of respondents reported unusually high portions of requirements that are vague and untestable, and requirement that were changed. An analysis shows seven respondents answered high values (indicating low quality), for both questions. These respondents all had at least a few years of experience, but most more than ten years. Other answers were analysed, they only shared one thing: they held slightly shorter workshops, often only lasting around 2 hours. It seems unlikely that this is the primary reason that caused them to experience lower quality requirements. It might be interesting to study the duration of workshops and result quality further.

Question 5: Do workshops increase productiveness of requirement development?
Surprisingly, the faster time-to-market (higher productivity, shortened duration) goal of workshops did not register as important with survey participants. Only 48% indicated workshops reduce the requirement effort. Interviews indicated that a major benefit was that it saves a lot of time going back and forward between stakeholders. We hypothesize that once workshops are used extensively, time saving aspects are less noticeable and therefore seen as less important than the increased soft results of a shared vision and commitment.

6.2.2 Validity

Were the right questions asked, and were the questions asked right?
To make sure the right questions were asked, the questions were based on extensive literature study, and survey theory. Furthermore, the questions were verified during brainstorm sessions with a requirement engineer. To test our survey, a trial survey was created in which five workshop facilitators were invited to complete the survey and give feedback. Three responses with feedback were received. Based on analysis we made some final changes.

Did we get the right participants?
To make sure the right people were invited, the survey participants were screened for having experience in facilitating requirement workshops prior to receiving the survey invitation. This screening was often done over telephone, but also in person, or in some situations by recommendation.

Did we get a representative number of participants?
The total number of participants was 28. These respondents are only a small sample and cannot be seen as representative of the complete field of requirement workshops. Nevertheless, the respondents are all experienced facilitators, from a variety of organisations. The results do give an indication about real practices.

Was the interpretation of results correct?
To verify and validate our findings and interpretation, relevant chapters of this research project have been sent to experts for feedback. We received feedback from seven experts with backgrounds in requirement workshops, requirement engineering, collaboration, and facilitation. Furthermore, only one project was found that also surveyed about the use of workshops in practice. We compared our findings to this project's findings in the analysis section.

6.3 Recommendations for Improvement

We would suggest using less questions on a more specific topic. The required 30 to 45 minutes was a large investment for participants. We learned that requirement workshops of different type exist, whereas the survey did not differentiate between types of workshops. More meaningful data might have been found if these types would have been integrated in the survey.

Analysis of our survey suggests further research is required in order to investigate how much time is spend on requirements, workshops, preparation and follow-up. Also it would be interesting to investigate what influences the wide variety of different answers about preparation and follow-up effort, and how respondents view the quality of the requirement that were created with workshops.

Chapter 7

Collecting Empirical Data

To find evidence of increased requirement quality and increased productivity, we intended to collect empirical data to measure requirement quality and the productivity of requirement workshops. While we failed to collect significant data, interesting observations were made.

Section 1 discusses our measurement design and its rationale. In section 2 we analyse the problems found while trying to collect empirical data, and re-evaluate prior research in light of our findings. In section 3 we discuss alternative plans we tried to use to collect data. Section 4 lists ideas for improvement of future requirement measurement plans.

7.1 Measurement Design

The model of our measurement design, using the Goal-Question-Metric (GQM) approach (Basili et al., 1994), is shown in Figure 7.1

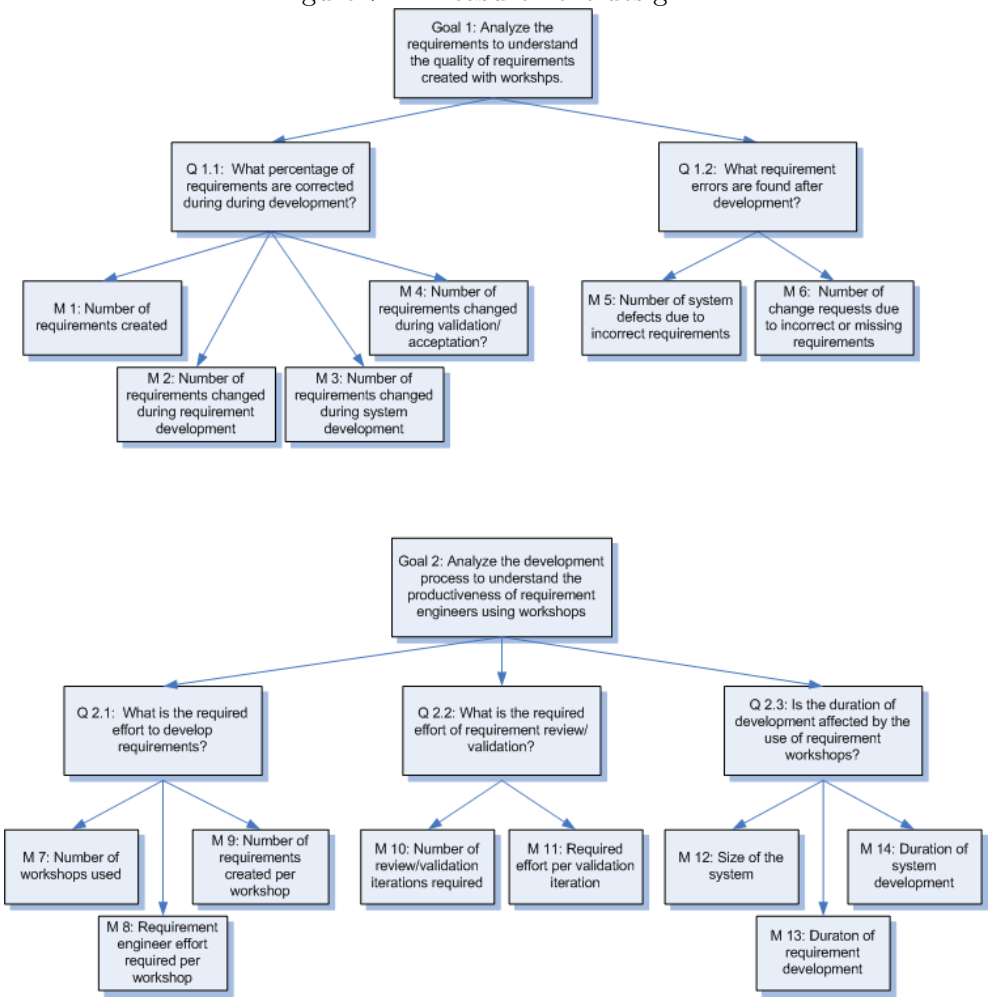
7.1.1 Rationale

Requirement Quality

Based on literature study, we found that the way requirement quality is measured is based on quality criteria of being accurate, precise, complete, concise, relevant, creative, consistent and feasible. A studied example of a project measuring requirements with these quality criteria is Duggan and Thachenkary (2003). This method can measure whether requirements are stated correctly, it does not however measure whether the correct requirements are found.

To test the quality of the requirement *process* with workshops, we judge quality based on the criteria of acceptance as measured by the changes that

Figure 7.1: Measurement design.



are made to the requirements. We hypothesize that if requirements are of high quality, they are correct as judged by the user and requirements will not be changed. A change to requirement is defined as either a change to, an addition of, or a removal of a requirement. To measure this, we needed to collect requirement volatility metrics.

We found only one requirement volatility measurement solution. Jones (2000) bases his volatility findings on requirement creep. Requirement creep is the growing scope of a project. Jones measures this by keeping metrics on total project size by counting its function points. This is not however an indicator of requirements churn. Requirement churn measures the changes made to requirements that do not increase function point totals. With good requirement change management, large changes could be made to requirements without it increasing the function point total. As an indicator of quality, requirement churn needs to be measured.

We used two sets of changes to measure requirement churn:

- Changes during system development. These are indicative of internal conflicts, and incomplete and incorrect statement of requirements. Errors found in requirements at the end of requirement process, based on validation and acceptation reviews, are indicative of a process that failed to find the correct requirements and identify conflicts.
- Problems that are found when the system is in use. Requirements could be stated correctly and be accepted by the user, but the resulting system could still not solve the problem. Problems that are found when the product is in use are an indicator of the invalidity of the requirements. They are indicative of missed stakeholders, incorrect project scope, and of a process that lead to the wrong solution.

Incorrect requirements of a system in use can be found in two ways: by analysing bugs and system defect reports, and by analysing change requests. Care should be made not to include defects by programming errors, and change requests due to the changing nature of organisations.

To analyse whether a change in the organisation caused the change request, project scope should be investigated. When change requests are in the scope of the development project, they count. When change requests are out of the original scope, it should be judged whether they are indeed out of scope for the project, or whether it should have been included in the project's scope originally.

Requirement Engineering Productiveness

To measure the productiveness of a requirement engineer, we looked at invested effort, and the duration of activities. The effort required to develop

requirements is dependent on the effort required to develop and the effort to verify, correct and validate them. To create a productiveness indicator we also needed to capture dimensions of both the workshops, the number of requirements, and the size of the system.

We hypothesize that when the productivity of the requirement engineer goes up, the duration of requirement development goes down. We test this hypothesis by measuring the duration of requirement development.

Finally, we hypothesize that when all requirement problems are found during requirement development, and clear and complete requirements are available, less problems will surface during development and development reviews. This should shorten the duration of development as well. To test this hypothesis, we included measuring the development phase.

7.2 Problems Experienced

Of 31 contacted organisations that used workshops, only two indicated they could and wanted to help. Three projects were measured in two organisations, and partial results were found. Data collected from one project is shown in Appendix D on page 99.

Several issues where found that prohibited collecting empirical data:

1. Organisations were hesitant to participate in this project due to the private nature of information required.
2. Organisations gathered requirements for a third party that did not want to cooperate.
3. Contacted organisations did not keep adequate records of change requests and changes made to requirements.
4. Logged changes often missed a categorisation or rationale.
5. Requirement statements were not often created during workshops, but stated afterwards.
6. Organisations rarely kept adequate effort metrics of requirement activities.
7. A small data set would not give meaningful data.

Recommendations for improvement and solutions are made in Section 7.4.

7.2.1 Re-analysis of Prior Research

Other research projects also investigated requirement workshops. In light of our findings, we re-analysed how other research projects dealt with these problems. Purvis and Sambamurthy (1997) investigated the perception of

JAD, not the productivity or quality of requirements itself, using a questionnaire survey. Davidson (1999) based their findings on interviews. They interviewed 34 people about experiences on 20 projects. Duggan and Thachenkary (2003) looked at the quality of requirements using two different styles of facilitation. They side stepped the comparing problem by holding laboratory experiments, and judged the resulting requirements on other requirement quality aspects. Maiden et al. (2004) did a case study on only one project, side stepping the problem of comparing.

Of all research projects found, Schalken et al. (2004) compared to this research project most. Schalken et al. looked at the effects of workshops, choosing to compare time spend on requirements between two methods in one organisation. The DSDM/workshop method was integrated into the organisation recently when the comparative study took place. Schalken analysed the project database, looking at invested time per function point. Satisfaction numbers came from workshop evaluation surveys. Schalken did not look at at the quality of requirements, types of workshops, workshop strategies and facilitation.

7.3 Alternative Plans

As it became clear we could not collect required data, we created three alternative plans: to extract required data from requirement management tools, to look at data collected by others, and to execute an experiment with a leading project planning and estimation tool.

7.3.1 Requirement Management Tools

Two leading manufacturers of requirement management tools were contacted. A requirement management tools records, structures and manages requirements in projects, logging changes, providing overviews, and tracing the origins of information to aid in analysis.

One manufacturer indicated it could not provide the requested information. The other manufacturer did help, and its management tool was evaluated. It did log changes, but not in a way suitable for generating meaningful data about requirement volatility. It could not generate a useful reports on changes to requirements that could be analysed.

One reason that caused the requirement management tool not to be able to generate useful reports was that it did not constructively log reasons, rationale, or type of change. Free-text entry did allow for some information, but no categorisation could be made about changes. Every small change to every aspect of a use case or requirement statement needed to be inspected

manually and visually to gather data. Even if the rational and reasons could be found by visually inspecting changes, the relatively short time frame of this project meant this was not a viable option.

7.3.2 Existing Data Collections

Several existing research initiatives were assessed to see if they could provide information. Based on research by Barry Boehm, we assessed both "Software Engineering Economics" and Cocomo II for data about requirements and volatility. No relevant and reasonably current information could be found about volatility. Although the Cocomo II did hold information about project and requirement phase duration, it did not differentiate between with and without the use of the workshop techniques. Also, it did not hold data on developments past 1999, and it did not hold information about requirement quality or volatility.

Also, we assessed publications from Capers Jones. "Software assessments, benchmarks, and best practices" and "Estimating Software Costs: Bringing realism to estimating, second edition" were inspected for relevant information. Although Jones did discuss JAD and gave an indication on decreased requirement creep in both books, he does not provide information about requirement phase duration or about requirement churn.

7.3.3 The Shortened Requirement Phase Experiment

Based on literature study and the interviews, we found at least one requirement workshop approach that held the potential to shorten the duration of requirement gathering by increased intensity of holding workshops in multi-day sessions. Also, anecdotal evidence presented in literature suggests that time and money can be saved, and overall productivity can be increased.

To test this statement, we executed an experiment with a leading project estimation tool. An information system development project of 500 function points was estimated using QSM SLIM-Estimate. This estimation tool is based on experiences from well over 6000 projects, and holds data about recent projects and methodologies. (QSM, 2007)

First, a normal scenario was estimated with the optimal balance between duration and effort. Second, the duration of the requirement phase (the Inception phase) was halved. We hypothesized that effort would not decrease as the workshops required increased levels of facilitation. A fixed time scenario was simulated, no changes where made to the final deadline.

Results include a projected 25% increase in overall project productivity. Furthermore we found a 30% decrease in required peak staff during construction

phase. The benefits resulted in an overall cost saving of 24%. Based on this experiment it can be concluded that productivity can be improved, and money can be saved by increased productivity in the requirement phase. This experiment is discussed in more detail in Appendix E on page 103.

7.4 Recommendations for Improvement

Several problems were found while trying to measure requirement quality and productivity. Several recommendations for improvement were found.

Problem 1: Organisations were hesitant to participate in this project, often due to the private nature of information required.
Requirements are sensitive information. Perhaps if more time was available, and more effort could have been spent on these contacts, it might have been possible to get access to requirements. This requires more time than we had available with this project. The set of statistics we intended to gather also was rather large, which we hypothesize also deterred some organisations from cooperating. A smaller more focused set of metrics would help.

Problem 2: Organisations gathered requirements for a third party that did not want to cooperate.
This research project was executed at the university. This had the benefit of being neutral which helped in opening some doors and make external contact to explore the subject. A downside was that we just did not have easy access to projects. One solution would be to execute a more focused research project at a requirement engineering organisation.

As stated above, an alternative would be to use a smaller set of metrics and spend more time with individual organisations to get more familiar and better emerged in the organisation.

Problem 3: Contacted organisations did not keep adequate records of change requests and changes made to requirements.
One way around this would be to analyse bug reports and change requests. Project administrations more often keep records of bugs and change requests than other requirement aspects we looked at. These metrics are contaminated by the changing nature of organisations and the quality of a project's test and validation process. Adequate analysis can give a valid indication of requirement quality.

Another option is used by Jones (2000), he uses the changing number of function points as a basis for making claims about requirement creep. (the continual growth of the total number of requirements) This method is another indicator of requirement change.

Problem 4: Logged requirement changes often missed a categorisation or rationale.

Both in contacted organisations and in the requirement management systems, the rationale for changes was missing. We found that the rationale can be found in project management tools rather than requirement management tools. Project management tools initiate changes at higher project levels, for which more often a rationale is given. It would be interesting to look at changes between two sets of requirements baselines, one before and one after the change. Together with the rationale for the change itself, this could give relevant data.

We learned that changes to requirements during the really early forming and brainstorming stages of the requirements should not be seen an indicator of bad requirement quality. In fact, workshops provide a way to detect problems by putting stakeholders together. These changes are indicative of increased requirement quality. It is only after the original set has been baselined that requirement changes are indicative of lower quality of requirements.

Project management would only start logging changes to the first requirements baseline. This means that the early changes that increase the quality of requirements are not counted as negative quality indicator. The data in project management systems can contain both the rationale for a change and, combined with version control systems, can also contain the changes made to baselined sets of requirements. This would be an interesting approach for future studies.

Problem 5: Requirement statements were not often created during workshops, but stated after the fact.
As requirements are often stated outside of workshops, productivity metrics about the actual number of requirements made during a workshop have little value. The number of requirements that were created with deskwork afterwards would not be a good indicator of workshop productivity, as it measures the requirement engineer's abilities more than the workshop's effectiveness. Should this kind of information be required, then researchers should get involved in the projects from the start to make sure usable metrics are gathered.

Problem 6: Organisations rarely keep adequate effort metrics of requirement activities.
Another improvement would be to look at a smaller set of metrics. For example only the duration of project phases. Project administrations more often keep record of this aspect than the other aspects. We found that organisations we contacted rarely measured the effort involved during the creation of requirements. If individuals did keep records, they often only included effort of facilitators or requirement engineers and not other stakeholders.

Problem 7: A small data set would not give meaningful data.
We learned that the type of facilitation and collaboration largely determines

the effectiveness and product quality of the workshop. Even small changes in the introduction of a session can create differences in workshop effectiveness and efficiency. When only a small set of data was found, it would not have given meaningful data. One solution would be to get a larger data set.

Another solution would be to also measure contributing factors. As explained in Chapter 8, many factors contribute to the result of a workshop. These factors could all be measured. One such factors is for example whether or not a team (including its facilitator) is balanced. This could be measured using a combination of Belbin and the Myers-Briggs Type Indicator tests.

Another approach is to use the tools created by the research field of collaboration engineering. It would be interesting to catalogue the activities of workshops for requirement development using the tools and patterns of the collaboration engineering research field. Using this pattern language would help in making more accurate statements about productivity of individual workshop activities.

Chapter 8

Requirements Workshops

This chapter is the first chapter that answers the research questions. This chapter focuses on *Question 1: What are requirement workshops?* and *Question 2: How are requirement workshops used?* It also helps answering *Question 3: Do workshops involve the user?* by giving an impression on how the user participates.

The information in this chapter is based on our literature study, validated and combined with results from interviews and the survey. It is therefore based on impressions about workshops, not on empirical data of studied projects.

8.1 Introduction

In 1977, Chuck Morris of IBM needed to gather requirements for creating a process design and screen designs. Based on the book 'How to Make Meetings Work', (Straus and Doyle, 1976) an innovative plan was created to get the users together with system developers and design the new process and screens. Joint Application Development (JAD), as the method was called, uses group creativity and dynamics to create the requirements for a system, including system objectives, screen design and report design. (Wood and Silver, 1989; Rush, 2006; August, 1991)

The process revolves around a series of cross-functional meetings, called facilitated workshops. "A facilitated workshop is a structured approach to ensure that a group of people can reach a predetermined objective in a compressed timeframe, supported by an impartial facilitator." (DSDM Consortium, 2003) Workshops often use low-tech visual aids such as flip-charts, brown paper, whiteboards, sticky notes and stickers.

The use of the word workshop is spreading. Workshops are used for almost anything that involves groups of people. There are even tango dancing

and poker playing workshops. These workshops usually consist of a short training course taught by the workshop leader, the expert and teacher. Facilitated workshops put the creativity of the participants at the center. An atmosphere is created "that makes participants able to empower each other to achieve brilliant objectives." (Hoogenboom et al., 2004)

Requirements workshops resemble the original JAD sessions. Participants "quickly and efficiently define, create, refine, prioritise and reach closure on deliverables (such as models and documents) that represent user requirements." Often held at the beginning of a project, they build positive, productive working relationships. (Gottesdiener, 2002, 2005)

The concepts and techniques of the original JAD method have been adopted by other methodologies and are used often. Jones (2000) indicates that 70% of inspected projects in excess of 100 function points used workshops for gathering and analyzing requirements. A non-exhaustive list of methodologies using workshops is shown in Table 8.1. Appendix A.3 on page 84 contains more information about the creation of JAD.

Table 8.1: Methodologies using workshops.

Name	**Reference**
Joint Application Development	Wood and Silver (1995)
Participatory Design	Bødker et al. (2004)
Rapid Application Development	Martin (1991)
Dynamic Systems Development Method	DSDM-Consortium (2007)

8.2 Benefits and Caveats

Based on literature study, the benefits of using workshops are: (Wood and Silver, 1995; Gottesdiener, 2002; DSDM Consortium, 2003; Carmel et al., 1993)

- Accelerate system design.
- Improve the quality of system design.
- Improve relations between business domain and IT.
- Increase stakeholder commitment and buy-in.

Workshops can provide benefits on projects that: (Wood and Silver, 1989; Gottesdiener, 2002)

- Have unknown or complex requirements.
- Have stakeholders of multiple different departments and backgrounds.
- Have a need for speed, requirement are required in a hurry.
- Have requirements that are visible to the end-users.
- Have a well defined sponsor for commitment to the process and ensures the right players attend and participate.
- Allows for enough time for planning and pre-work to prevent workshop catastrophes.
- Have an experienced and neutral facilitator to offset dysfunctional group behaviour.
- Can have all participants being equal during the workshop, to prevent follow-the-leader behaviour.
- Have participants with mandate to make decisions, preventing "Well, let's run this by Marketing". Marketing should be right there.

Workshops however, "are like black-holes, workshops have their own laws of physics, which sometimes differs from the standards we are used to." (Hoogenboom et al., 2004)

Garner (1995) wrote:

> This JAD stuff is not intuitive to the untrained. And therein lies its danger. Proceed blindly into that quicksand of good intentions and you face the morass of miscommunication, misplaced expectations and skyrocketing costs. If that does not give you pause, perhaps this will. Getting started is the hardest, most brain-numbing exercise and the most important of them all.

8.3 Facilitation

To attain the benefits of workshops, and prevent counterproductive group behaviour, the workshops technique makes use of a skilled leader, called a facilitator.

Group facilitation is the process "in which a person whose selection is acceptable to all members of the group, is substantively neutral, and has no substantive decisionmaking authority diagnoses and intervenes to help a group improve how it identifies and solves problems and makes decisions, to increase the groups effectiveness." (Schuman and IAF, 2005)

McConnell (1996) indicated that when workshops fail, it is almost always because of the facilitator. The session leader "needs a rare combination of skills. The leader must have excellent communication and mediation skills. The leader must mediate political disputes, power struggles and personality clashes."

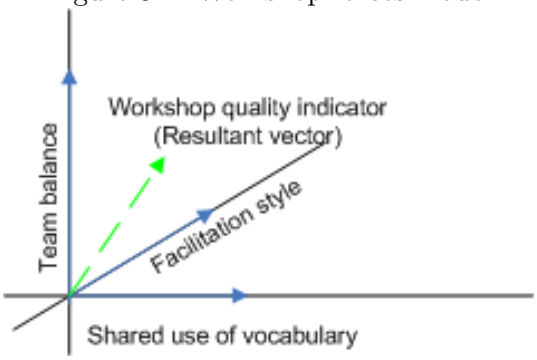
Hoogenboom et al. (2004) list five facilitator types. The friendly facilitator is so friendly and forthcoming that the participants do what he/she requests. The host facilitator prepares and manages the process, environment and facilities in such a way that it is a joy to do the work. The director facilitator is about orchestration, and getting a group to work together on something that might not be needed by the individual participants themselves but is required by the organisation. The ambassador facilitator is almost invisible to the group, but things 'just happen' to fall in place and work out for the best. Hoogenboom et al. also lists the manipulative facilitator, but state you should not make use of manipulation, if you must then do it out in the open.

Many training programs exist to train facilitation skills. The International Association of Facilitators (IAF) created a certification process in which facilitators are assessed. If the candidate shows the required competencies, he or she can use the 'Certified Professional Facilitator' (CPF) designation. The IAF lists competencies in the following categories: (IAF, 2004)

- Create collaborative client relationships.
- Plan appropriate group processes.
- Create and sustain a participatory environment.
- Guide group to appropriate and useful outcomes.
- Build and maintain professional knowledge.
- Model positive professional attitude.

Not all workshops require the same level of preparation and facilitation. Using an existing collaboration design or targeted training, it might not be necessary to hire an expensive professional facilitator. (Briggs et al., 2001; Duggan and Thachenkary, 2003) Whether a professional facilitator is required or not, preparation and a workshop plan can prevent getting the wrong results, both hard and soft, and prevent much participant frustration.

Figure 8.1: Workshop forces model.



8.4 Preparation

To prevent unproductive sessions, a good workshop plan must be made. A good plan can prevent getting the wrong results. Results include both hard results: the ideas, plans and requirements created; and soft results: the satisfaction, acceptance, and commitment of the individual and the group to the hard results. (Hoogenboom et al., 2004)

Workshops are more sensitive to bad preparation than interviews. A badly prepared interview might go unnoticed to the interviewee, it will not go unnoticed in a workshop. As our survey indicated that often 12 participants attend a workshops, the impact of a being unprepared multiplies by 12, not including the group process that might escalate any problem even more.

A rule of thumb is suggested to spend two to three times as much effort on preparation, and two times as much on follow-up compared to workshop duration. (Hoogenboom et al., 2004) It was found in our survey that a 2:1 effort of preparation and 1.5:1 effort of follow-up was required over run-time. For an 8 hour workshop, respondents required on average 15 hours of preparation and 10 hours of follow-up.

After the workshops has ended participants often remember the location, their personal fun-factor, and perhaps some other observations. There are however multiple factors that contribute to the process and outcome of a workshop. Examples are: the agenda; the style of facilitation; the setup and use of the collaboration techniques; the participants and whether or not they have complementary skills, are balanced, and have jelled; and whether the group uses words that hold the same meaning.

To illustrate, Figure 8.1 plots three factors as vectors in an x,y,z graph. This shows the workshop outcome as a combined result of these vectors. Good workshop preparation is crucial to make sure a plan is created that shapes all these factors.

8.4.1 Four Steps in Preparation

The preparation of the workshop involves four steps: Orientation, Planning, Pre-Workshop Session, and Logistics.

Orientation: the facilitator starts by holding interviews to get an idea about the workshop's goals and attendants. Stakeholders are analysed to familiarise which backgrounds and viewpoints of each of the stakeholders, to make sure the right people will be invited to the workshop. Often a working document is created holding suspected requirements, open issues, and assumptions based on the interviews. The document also contains the workshop agenda, a tentative list of participants, and a workshop plan or script.

Planning: A plan is created on three levels. First, at the What level, decide on the goals of the collaboration and the steps the group must take to reach the goals. Second, at the How level, a plan must be made how the participants move through each of the steps. These collaboration processes are often called building blocks or thinkLets. Third, at the Activity level, for each activity a process with rules must be designed, and a script introducing and ending the activity. Special tools must be prepared for as well, e.g. configuration of group support systems, and templates or predefined lists of quality criteria. (de Vreede and Briggs, 2005; Kolfschoten et al., 2006; Briggs et al., 2001; Hoogenboom et al., 2004) Care should be made to invite the appropriate participants. The right participants are often hard to get, and they will not enjoy workshops when they cannot contribute and waste their time.

Pre-Workshop Session: With larger workshops participants come together in a pre-workshop session to discuss the plans for the workshop. This session is held if participants have to come to the main workshop prepared. Participant preparation can include reading relevant documentation, doing individual brainstorms on subjects like problems, goals, relevant business rules, a process description, and an early set of requirements or usage scenarios.

Logistics: Preparation includes making sure a good room is booked and available. Make sure the room is configured correctly before the workshop starts, make sure whiteboard markers, flip-charts, brown-paper, stickers, coloured sticky-notes, and sticky tape is available. A computer with beamer might be very useful for demonstration purposes. Make sure there is something to eat and drink. (Gottesdiener, 2002; Wood and Silver, 1995)

8.5 Workshops in the Software Development Process

Requirements workshops are often held in a series of sessions, lasting 3 or 4 hours each. These sessions are held two or three times a week, and on average between 4 and 12 participants attend a workshop. This includes both the IT domain and business domain. Before workshops are held the analyst, often a requirement engineer, holds interviews to get familiar with the business domain, politics, the problem at hand, and project goals.

Based on interviews, two workshop methods have been found: The Collect, Consolidate and Confirm method and the Why-What-How method.

8.5.1 The Collect, Consolidate and Confirm Method

The Collect, Consolidate and Confirm method consists of three types of activities, of which two are often done with workshops. The first type of activity is used to find and collect facts and information with workshops. These workshops are entered largely blank, and all information relevant to the subject is collected. These workshop are used as a source of information. They promote mutual learning, and creating clarity and a shared vision.

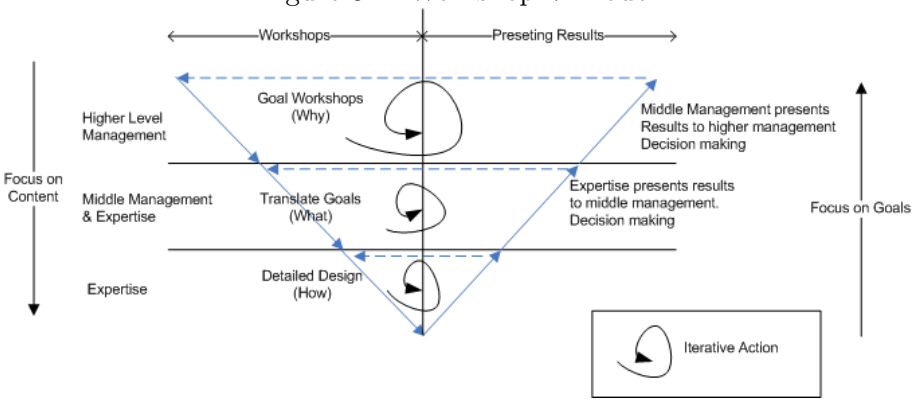
The second type of activity is the consolidation step often not executed with workshops. Information found in the collect workshops is processed and refined for further use. This information can be used as input for creating diagrams, models, requirements, use cases, and even prototypes. This can be done in workshops but, as this activity often requires more time than is available in workshops, is more often done as deskwork.

The third type is used for confirmation and validation. These workshops are used for verification, validation, and getting agreement. This type can be seen as decision making. These sessions use presentations, show protypes, and ask for feedback. These sessions can also be used to test models, requirements and use cases, the results of the collect workshops and consolidation activities.

Workshops can comprise of all three types of activities. For example a brainstorming sessions start with collecting ideas, then consolidates by grouping ideas, and creating potential solutions, which are then confirmed and validated.

Some projects only use the confirm workshop to promote a plan, often created outside of the workshop without user participation or involvement. In this scenario workshops are used to educate users about plans, ‘sell’ ideas, and get user approval without making too many changes to the original ideas.

Figure 8.2: Workshop V-Model



8.5.2 The Why-What-How Method

The Why-What-How method follows the Requirement Pyramid model as shown in Figure 2.2 on page 6. Three different workshops can be found: Why-Workshops, What-Workshops and How-Workshops. The workshops start with a planning workshop for higher level management, as shown in Figure 8.2.

First Why-Workshops are held to clarify why a system is needed, define goals, and set project scope. Then What-Workshops are held to translate goals to user tasks. Finally, How-Workshops are used to specify and design a system that supports the user in executing their tasks and subject matter experts and end-users take decision about specifications and product design. At the this level, workshops tend to use less diverse groups in less formal group meetings (or even interviews) to get the details and design prototype screens and reports. Table 8.2 describes these workshops in more detail.

Multiple workshops of each type could be held in an iterative style. Also, on each level the Collect, Consolidate and Confirm method can be used. Once finished, results are presented in some form to higher levels, often as defined in project management methods like Prince2, using presentations, or inviting executives to the closing stages of the workshops. If stakeholders are not satisfied with results, more iterations can be made following the dashed lines in Figure 8.2.

Often fewer higher level workshops are held than expertise design workshops. This matches the fact that significantly more system requirements are found and required than business requirements. Often per increment, a different set of design workshops is held.

Table 8.2: The Why-What-How Workshops.

Why-Workshops	
Goals	Find out why a system is needed
Activities	Analysing organisational needs, defining goals, problem solving, creating vision, making context diagrams models, identifying high-level system requirements, defining a system scope, planning the following workshops.
Participants	Executives, members of steering committees.
What-Workshops	
Goals	Define what is needed for the user to fulfill organisational needs and goals.
Activities	Translating general ideas into actual solutions, focusing on the steps required to achieve the goals, involving users and their tasks. Domain models, process models and use cases are discovered and elaborated. System quality attributes can be found using quality attribute workshop techniques. (Barbacci et al., 2003) Stakeholders analysis to create a participant list for the How-Workshops.
Participants	Project management, executive middle management, business analysts, requirement engineers, expert end-users, IT architects.
How-Workshops	
Goals	Define how a user executes his tasks, and how the system supports these tasks.
Activities	Activities include detailed design and specification, prototyping, reviews of screens and reports, and specifying the system's requirements.
Participants	Subject matter experts and end-users.

Chapter 9

Attainability of Workshop Benefits

This chapter answers the following research questions:
Question 3: Do workshops involve the user?
Question 4: Do workshops increase requirement quality?
Question 5: Do workshops increase productiveness of requirement development?

This chapter presents an analysis of the primary benefits and issues of workshops, based on literature study, interviews, the survey, and introspection, combined with data from relevant prior research projects. It is based on perceptions and opinions of experts, it is not based on empirical data about studied workshopped projects themselves.

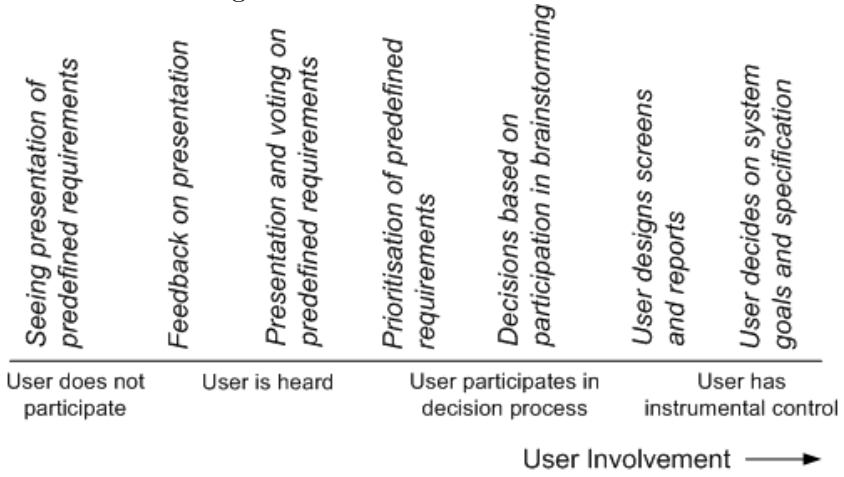
In section 1 we look at the aspect of user involvement in design. In section 2 we argue that requirement workshops increase quality despite research results implying otherwise. Section 3 presents an analysis of how and where resources can be saved. Section 4 holds a discussion on the seeming paradox of using the workshop as process improvement.

9.1 Workshops Involve the User

The number one reported success factor for software development projects is user involvement (Standish Group, 2000) The primary selling point of using workshops is that it involves the user in the design of software systems.

Involvement is a subjective, psychological state. The level of involvement is affected by three interdependent factors: hands-on participation, the relationship between user and designer, and the user's level of responsibility. (Hartwick et al., 1994)

Figure 9.1: User Involvement Scale



First, hands-on participation. Survey respondents report that, in practice, the techniques used most often in workshops were prototypes, use cases, and scenarios and stories. These are all techniques that revolve around the user. Based on these findings, we conclude that users do indeed have hands-on participation in system specification in workshops.

Second, the relationship between user and designer. Workshops can improve the relationship between users and designers. Based on fundamental research Purvis and Sambamurthy (1997) report that both users and designers found JAD workshops facilitated higher quality user-designer interactions, partnerships, and improved communication.

Third, increased level of responsibility. To experience increased levels of responsibility users should play an active role, and should at least be heard but preferably have instrumental control over the decision making process for attaining the benefits of user involvement. (Hunton and Beeler, 1997) Hartwick et al. (1994) state that the level of responsibility can be affected by hands-on activity and a good user-IT relationship. We concluded workshops promote hands-on activity, and discovered prior research that found relationships improved with the use of workshops. Whether or not the user has instrumental controle over system specification, and whether this would even be wanted, depends largely on the project and design of the workshop.

Different activities in workshops score differently on the scale of involving the user during the workshop. As an indication, an example scale of activities is shown in Figure 9.1. The actual placement of the activity depends on the design of the activity and in the execution of it. The facilitator's choices and personal style move the placement of activities over the axis.

Most decision processes have winners and losers. When participants have

real input and their concerns are heard and weighed, participants will generally be more satisfied with the process and its results. McKeen et al. (1994) found proof that true user participation, user influence and user-developer communication was positively related to user satisfaction. Participation was also directly linked to a user's satisfaction with systems development. The participation-satisfaction link was also found by Mohammed and Ringseis (2001) and Witt et al. (2000).

To get users involved, facilitators have the responsibility to choose the correct processes to actively use the creativity of users in problem solving, have the user participate in the design process, and actively work on the relationship between IT and user participants. We have shown using the three factors of involvement that workshops can get the users involved.

9.2 Workshops Increase Quality

Requirement engineering has a large component of problem solving, and workshops are used to solve problems with the participation of the users and other stakeholders. Experiments with ideation (generating ideas) show that group brainstorming, often used in problem solving, shows decreased productivity and quality compared with combining ideas generated by individuals. (see Sutton and Hargadon (1996); Nijstad and Stroebe (2006) for a listing of citations)

Productivity loss in group brainstorming is contributed to production blocking (waiting on each other), free-riding (not participating) and evaluation apprehension (apprehension of being judged) (Diehl and Stroebe, 1987) Besides being unproductive, groups can show dysfunctional behaviour like group conforming behaviour, individual members dominating the process, slowing down the process by digressing from the main agenda and arriving at a group decision that is contrary to the desires of its individual members. (Duggan and Thachenkary, 2003)

Katzenbach and Smith (1993) define a team as "a small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable." Most productivity studies were done with participants that did not have past or future relationships, where the ideas generated where not used, where participants did not have skills that complement other participants, and participants did not have expertise in brainstorming and leading brainstorming sessions. (Sutton and Hargadon, 1996) In short, research projects have often used homogeneous groups instead of cross-functional teams. Although the participants in requirements workshops are not necessarily teams, they do show many of the attributes of a team. These differing attributes are important to the group process and its results.

The magnitude of software development means problems often cannot be solved by individuals and need the input of people with different backgrounds and expertise. It is in the combination and collaboration of diverse participants in teams that the advantages become visible. Few research projects researched cross-functional teams, but fundamental research *has* shown that these teams, teams that have diversity among their members, do improve decision quality (Jackson et al., 1995) and do lead to faster product development. (Eisenhardt and Tabrizi, 1995)

Furthermore, facilitators can prevent causes of productivity loss and increase the quality of results. Process facilitation in workshops has a positive impact on the process and increases participant satisfaction. (Miranda and Bostrom, 1999) In software development projects, process facilitation significantly increased productivity. (Unger and Walker, 1977) Also, team diversity during brainstorming diversifies cognitive stimulation, the process in which new thoughts are triggered by seeing the ideas of others. This further increases productivity in cross-functional sessions. (Nijstad and Stroebe, 2006)

There is more to brainstorming than productivity. Most experimenters use the number of ideas as the sole effectiveness outcome. (Sutton and Hargadon, 1996) The results of group sessions are both hard results (in this case the number of ideas) and soft results, the satisfaction, acceptance, and commitment of the individual and group to the generated hard results. (Hoogenboom et al., 2004) It is in the soft results that important benefits can be found.

Participants in cross-functional sessions are more satisfied with their own performance. They enjoy the process and their performance in it more than brainstorming individuals. (Nijstad and Stroebe, 2006) When diverse groups come to a consensus, participants also show increased satisfaction with the results. (Mohammed and Ringseis, 2001) The satisfaction with process and results is of crucial importance for commitment to the created solution.

Individuals often cannot solve the issues faced in the growing complexity of software projects. Teams can be dysfunctional during brainstorming, but these problems can often be mitigated by facilitation. The quality of group work should be measured in both the hard results and soft results. Soft results quality is increased: the way cross-functional teams solve problems increases satisfaction with the process and its results, increasing commitment to the created solution. The quality of hard results are increased by combining ideas from people with different backgrounds, and, with correct facilitation, creating more and better quality ideas. (Duggan and Thachenkary, 2003; Reinig et al., 2007)

Requirement workshop facilitators indicated in our survey, that if they could not use workshops, either the quality of requirements would be lower due to

decreased communication between, and input of, stakeholders, or it would take more time to reach the same level of quality.

Despite potential problems, there is a clear case for cross-functional teams increasing quality of hard and soft results. Based on numerous software project assessments, Jones (2000) reports that the workshops approach show increased quality as they decrease requirement volatility over the project's duration by two thirds.

9.3 Workshops Increase Requirement Engineer Productiveness

Workshops are said to reduce the time needed for the generation of requirements. Reported benefits include a 30-40% time reduction in design and 20-30% in implementation, more than halving the effort per unit of work, and saving hundreds of thousands of dollars in a single project. (Carmel et al., 1993) Is there any empirical evidence to back this up?

There are four areas in which workshops can save time over individual interviews. Workshops shorten the lines of communication, the perceived productivity loss of group conflicts in workshops cause benefits later on in the project, and cross-functional teams shorten the delay between steps. Finally, the usual series of workshops can be compressed in a few intensive days of collaboration.

With all stakeholders and decision makers present, there are short lines of communication in workshops. Stakeholders sit eyeball to eyeball which prevents going back-and-forth between these stakeholders to resolve conflicts. This can significantly save time in getting a shared view. (Wood and Silver, 1995) Interviews with facilitators indicated this is indeed seen as a big advantage of using workshops over interviews. The majority of respondents in our survey stated that using workshops reduced the required effort.

Cohen and Bailey (1997) hypothesize that the perceived productivity loss in group process of diverse teams is in part attributed to conflicts between participants. These conflicts are now found early in the development process, causing an early productivity loss but increased productivity during later stages, when making changes is more expensive in both time and money. While no research was found that tests this hypothesis, the majority of survey respondents indicated that putting together stakeholders with conflicting needs in workshops did not introduce a project risk.

Based on research, it was found that cross-functional teams lead to faster product development. Eisenhardt and Tabrizi (1995) discovered that when more diverse functions are involved in a team, the wait time between steps are reduced. As example Eisenhardt and Tabrizi indicate that the time

between design and prototype will likely be reduced when manufacturing is represented on the team.

Another method of saving time was found during this project. The Accelerated Solution approach revolves around a highly engineered, intensive, focused and facilitated series of collaborations in a special work environment. In two or three days of collaborative sessions often 70 or more workshops are held, which are otherwise often spread out over weeks or months.

An exploratory experiment with a market leading project planning and estimation tool was held to see the results of halving the duration of the requirement phase. The experiment projected a significant increase in productivity. Furthermore peak staff required during construction phase significantly decreased. These benefits resulted in an overall cost saving of 24%.

Two research projects found that using workshops in practice can indeed improve efficiency for software development projects in practice. Both Schalken et al. (2004) and Davidson (1999) report that projects that use workshops were found to be more productive than the traditional methods of interviews, but only under certain circumstances. Based on empirical evidence, Schalken et al. find that workshops as used by DSDM are more productive than traditional methods (using interviews) for projects over 171 function points.

We have indicated four areas in which time can be saved using workshops. Furthermore, we have discussed existing evidence where workshops were indeed found to be more productive. Finally, a method was discussed that holds the potential to significantly decrease requirement phase duration, and seen that the impact of this in a software development project can increase overall productiveness and save costs. We conclude that productivity can be improved, and money can be saved using workshops.

9.4 The Paradox of Introducing Improvement

Despite the stated reasons why workshops should lead to better results, prior research shows a different picture. A possible explanation for this is that introducing the workshop method into an existing organisation does not immediately produce improved results, and does not immediately result in the perception of improvement. This perception is seemingly validated by the required learning curve and by measuring the perceptions on benefits instead of measuring projects themselves.

We argue that as the roles of the stakeholders change, and the workshop technique is often adapted to fit the organisation and development methodology, it suppresses expected benefits at first. Without adequate action, this could continue to limit or even negate the potential benefits workshops offer.

Role of the User

Research about workshops found that from the user perspective acceptance and consensus management did not increase, while from the designer perspective it did. (Purvis and Sambamurthy, 1997) Another project also found that users did not indicate increased satisfaction with results of the newly introduced method over that of the traditional method. (Schalken et al., 2004)

Workshops changes the levels of responsibility and participation required from the user. Previously users were only interviewed, surveyed, or monitored while they did their work. Now users have to actively contribute to the creation of a system requiring the use of communication and negotiation skills, to learn about IT, and often spend more time to attend workshop sessions as well. Schalken et al. (2004) hypothesize that the increased participation also increases expectations about results, which means users are more likely to be disappointed with only smaller improvements.

Role of the Designer

Workshops require that developers have skills at planning, communicating, negotiating, and facilitating. We found all three research projects at least hinted at the fact that designers were less confident with the JAD method, and had less training and experience. Purvis and Sambamurthy (1997) stated assuming that the decreased levels of confidence must also affect users' perceptions of JAD and workshops.

The Role of Adaptation

Furthermore, Davidson (1999) reported that when workshops are introduced into an existing organisation or development methodology, they are often adapted to fit in, reducing its effectiveness. Examples of this are: shortening the JAD sessions, not assigning business area personnel to the project, involving surrogates instead of real users, not adequately training analysts, continuing to use difficult for users to understand models, performing key analytical activities outside of workshop, and so on.

As Davidson reported, success rates were low. Only 30% of projects reported better quality requirements, only 15% reported defining requirements more efficiently, 15% reported user satisfaction, 10% reported requirements were defined faster, and only 5% reported consensus on requirements. We would expect that adapting the workshop method to the existing organisation or development method would limit or even negate the potential benefits workshops have. This shows that in order to experience benefits from workshops, workshop constraints must be met.

Measuring Perceptions

The workshop method is significantly different from the traditional methods, and consist of different activities, different ways of solving problems, and require different skills, from both the business domain and the IT domain. When empirical studies measure the perception of a method, the real benefits are not actually found, only the impressions thereof. And impressions, as shown, could well be coloured by the new roles, new skills, and new experiences.

Based on perceptions, Davidson found improvements in 10-30% of the projects but seeing the difficulties would not advice on using it for larger projects. Whereas based on actual measured facts about projects, Schalken et al. advice that it works better for larger (171fp+) projects.

Based on analysis of Davidson's paper, he seems to have at least measured two different JAD methods. Purvis and Sambamurthy studied 57 different JAD projects. Furthermore, Schalken et al. stated measuring DSDM, yet another method. We found that many different types of workshops are held, workshops are used for many different goals and reasons, and that workshop results are influenced by many aspects. In light of these findings, it is unclear what exactly was researched by these projects. Their results are therefore difficult to interpret and explain.

Discussion

Apprehension for change, combined with the new roles and competencies, means that implementation of workshops as process improvement does not immediately increase stakeholder satisfaction and product quality. We hypothesize that the learning curve of workshops, combined with the required change in the organisation instead of changes to the workshop method, means that few immediate improvements can be found.

Studies about perception and people's experiences certainly have real value, and give an indication as to what really happens. But few research projects have focused on measuring workshop projects and requirements themselves, and more are required to make accurate statements about the effectiveness of workshops.

Chapter 10

Conclusions and Future Work

10.1 Conclusions

The goal of this project was to answer the question: *"Are workshops a valuable tool for requirement elicitation, or an added project risk?"* To answer this question a strategy of four approaches was used: literature study, gathering empirical data, a questionnaire survey, and interviews.

Question 1: What are requirement workshops?
Requirement workshops are guided sessions in which relevant stakeholders with diverse backgrounds and viewpoints sit together to collaborate on a variety of activities required to gather information, set goals, solve problems, and define and validate models, to create scenarios, requirements and designs that meet the organisation's and stakeholder's needs.

Question 2: How are requirement workshops used?
Based on our research, workshops are most often used to create or modify information systems. Workshops are often held in a series of sessions, rather than two or three day consecutive sessions. We found sessions often only lasted two to four hours, were held two or three times a week, and were often attended by four to twelve participants. Workshops are used more often for early requirement activities than for actual system specification. We found that specification of requirement statements and use cases was often done after the workshop, using information generated during the workshop.

Question 3: Do workshops involve the user?
Whether a user feels involved depends on how the user participates, the relationship between the stakeholders and the level of responsibility as experienced by the user. Requirement workshops have stakeholders participate in the specification of a system in a wide variety of activities. End-users

participate in the creation of scenarios and use cases, prototype and screen design, and prioritisation and validation activities.

For users to be involved, workshops also provide ingredients to improve relationships. Relations are to be improved by shortening the lines of communication, promoting mutual learning, and creating a shared vision based on a shared vocabulary. Depending on project specific attributes, facilitation plans, and whether or not the user actually attends the workshops, determines whether or not the user is involved.

Question 4: Do workshops increase requirement quality?
To find empirical evidence of the benefits of workshops, we tried collecting project data. We found no suitable way to measure requirements in literature, and created a new measurement design to collect the required data. We did not, however, find enough organisations that could provide the required data. While this project did not succeed in collecting the required empirical data, insights have been found that provided relevant information about answers, and new ideas that can be used to help succeed future projects.

The magnitude of current day projects means significant benefits can be found in the combination of backgrounds and viewpoints of stakeholders in cross-functional teams, to create a more balanced and complete solution. Workshops provide a way to execute these group elicitation tasks. Having the end-user participate in system design will help building a solution that more closely fits in with the end-user's tasks and activities. Besides the quality of the system specification, workshops can also provide improvement to another quality attribute. The process of collaboration in cross-functional teams can result in increased satisfaction, acceptance, and commitment of the participants to the generated solution.

Question 5: Do workshops increase productiveness of requirement development?
The exploratory nature of the interviews and the survey, combined with missing empirical data about projects, makes it difficult to draw conclusions on the attainability of benefits. Workshops do however shorten the lines of communication between the stakeholders and developers, preventing a lot of going back and forth, and "let's run this by..." delays.

Furthermore, when workshops are compressed in multiday sessions, like Accelerated Solutions workshops, phase duration can decrease. Also, workshops provide a way to detect and solve misunderstandings and conflicts between stakeholders, users and developers that may have remained hidden in a specification document. This can prevent problems and increase productivity in later project phases.

Constraints
There are constraints on workshop adoption, and using workshops introduces new risks. Most of these risks, such as inviting the wrong participants,

inadequate preparation, the right timing for the right project activities, insufficient follow-up, and so on, are equally applicable to interviews and other elicitation activities.

Workshops do require an appropriate project, a good agenda, skilled facilitation, a good collaboration design, intensive participation by real users, and the appropriate environment to execute the workshop. These constraints and risks make attaining benefits difficult. When these constraints are met however, and both users, executives and designers are getting more familiar and experienced with the method, more benefits should be reported more often.

Conclusion
Workshops are not a silver bullet solution and constraints and risks certainly exist. No project data about benefits could be collected, but based on interviews, analysis, and the survey we conclude that risks are often manageable, and that benefits outweigh the risks: Workshops are a valuable requirement elicitation tool.

10.2 Future Work

During the execution of this project, some ideas were created for future investigations into the productivity of requirement workshops.

In analysis of survey results, a wide variety in answers of required preparation time were found. Combined with a fairly even spread of activities and models that were used, this shows that workshops are used for a wide variety of tasks not just defining requirements. It would be interesting to catalogue the activities of workshops for requirement gathering using the tools and patterns of the collaboration engineering research field. This would help in making more accurate statements about productivity of workshop activities.

Many factors have been found that contribute to workshop success, including the facilitator, preparation, the collaboration design, the participants, and the balance of the team. It would be interesting to find out which of these factors influences the outcome the most, to discover which of these factors make or break a workshop. It would also be interesting to see if and when participants in requirement workshops show team characteristics, and whether that can be linked to increased (hard and soft) quality in projects.

We found that that project management tools might offer a better way of measuring requirement volatility than requirement management tools. Project management tools initiate changes at higher project levels, for which more often a rationale is given. Changes between baselined set of requirements can be used to measure changes for a given change request. Project

management tools often start logging change requests after the first requirements are baselined. This means that the early changes caused by workshops that increase the quality of requirements are not counted as negative quality indicator.

Finally, because of their focused activities and magnified attributes, the Accelerated Solution workshops are an interesting subject for further research. Do benefits outweigh the costs of these workshops as commercial based evidence suggests? Great effort is expended to guide participants through the phases in a short time; does the developed solution still have commitment once people return to the workplace? The answer to this question could well be applicable to the question of how far the requirement phase can be compressed before quality deteriorates.

Bibliography

A. Abran and J.W. Moore. *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, 2004.

Ghada Alaa and Jaco Appelman. Process Support for Agile Requirements Modeling and Maintenance of E-Projects. *Proceedings of the twelfth Americas Conferences on Information Systems, Acapulco, Mexico*, 2006.

J.H. August. *Joint application design: the group session approach to system design*. Yourdon Press Upper Saddle River, NJ, USA, 1991.

M.R. Barbacci, R. Ellison, A.J. Lattanze, J.A. Stafford, and C.B. Weinstock. *Quality Attribute Workshops (QAWs)*. Carnegie Mellon University, Software Engineering Institute, `http://www.sei.cmu.edu/publications/documents/03.reports/03tr016.html`, 2003.

V.R. Basili, G. Caldiera, and H.D. Rombach. The Goal Question Metric Approach. *Encyclopedia of Software Engineering*, 1:528–532, 1994.

K. Bødker, F. Kensing, and J. Simonsen. *Participatory It Design: Designing for Business and Workplace Realities*. Mit Press, 2004.

R.O. Briggs, G.J. De Vreede, JF Nunamaker Jr, and D. Tobey. ThinkLets: achieving predictable, repeatable patterns of group interaction with group support systems (GSS). *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, page 9, 2001.

Frederick P. Brooks. *The mythical man-month: Essays on software engineering*. Addison Wesley, 1995.

Capgemini. *Accelerated Solutions Environment: At the Intersection of Creativity and Collaboration*. Capgemini, `http://www.capgemini.com/collaboration/tools/ase/`, last visited 2007-08-10, 2007.

E. Carmel, R.D. Whitaker, and J.F. George. PD and joint application design: a transatlantic comparison. *Communications of the ACM*, 36(6): 40–48, 1993.

M.G. Christel and K.C. Kang. *Issues in Requirements Elicitation*. Carnegie Mellon University, Software Engineering Institute, 1992.

S.G. Cohen and D.E. Bailey. What Makes Teams Work: Group Effectiveness Research from the Shop Floor to the Executive Suite. *Journal of Management*, 23(3):239, 1997.

E.J. Davidson. Joint application design (JAD) in practice. *The Journal of Systems & Software*, 45(3):215–223, 1999.

A.M. Davis and A.M. Hickey. Requirements Researchers: Do We Practice What We Preach? *Requirements Engineering*, 7(2):107–111, 2002.

G.J. de Vreede and R.O. Briggs. Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks. *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 17c–17c, 2005.

M. Diehl and W. Stroebe. Productivity loss in idea-generating groups: toward a solution of the riddle. *Journal of Personality and Social Psychology*, 53:497–509, 1987.

Wim Dijkgraaf and Mike Van Spall. *Begin bij het eind met SMART requirements*. `http://www.synergio.nl`, 2007.

DSDM Consortium. *Framework for Business Centred Solutions Handbook*. DSDM Consortium, 2003.

DSDM-Consortium. *DSDM Manual*. DSDM Consortium, `http://www.dsdm.org/version4/2/public/`, last visited 2007-08-10, 2007.

E.W. Duggan and C.S. Thachenkary. Higher Quality Requirements: Supporting Joint Application Development with the Nominal Group Technique. *Information Technology and Management*, 4(4):391–408, 2003.

EBG Consulting. *Are these your requirements problems?* EBG Consulting, `www.ebgconsulting.com/reqtproblems.php`, last visited 2007-08-10, 2007.

K. Eisenhardt and B.N. Tabrizi. Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry. *Administrative Science Quarterly*, 40(1), 1995.

Donald Firesmith. *The Business Case for Requirement Engineering*. Carnegie Mellon, Software Engineering Institute, 2003.

Rochelle Garner. Why JAD Goes Bad. *Computerworld*, 28:87–88, April 1995.

R.L. Glass. *Facts and Fallacies of Software Engineering*. Addison-Wesley Boston, 2003.

E. Gottesdiener. *Requirements by Collaboration: Workshops for Defining Needs*. Addison-Wesley Professional, 2002.

E. Gottesdiener. *The Software Requirements Memory Jogger*. Addison Wesley Professional, 2005.

P. Grunbacher and R.O. Briggs. Surfacing tacit knowledge in requirements negotiation: experiences using EasyWinWin. *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, page 8, 2001.

J. Hartwick, H. Barki, B. Eg, et al. Explaining the role of user participation in information system use. *Management Science*, 40(4):440–465, 1994.

A.M. Hickey and A.M. Davis. Elicitation technique selection: how do experts do it? *Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International*, pages 169–178, 2003.

Mark C. Hoogenboom, Petra van Krugten-Elgersma, Birgit Klomps, and Tommes Snels. *Workshops: How to facilitate workshops in software engineering related environments*. Bergboek, Noord-Scharwoude, Netherlands, `http://www.kirjaboek.nl`, 2004.

J.E. Hunton and J.D. Beeler. Effects of user participation in systems development: a longitudinal field experiment. *MIS Quarterly*, 21(4):359–388, 1997.

Kimberly Hyde. *Brainstorming Information*. Case Western Reserve University, `http://www.case.edu/artsci/engl/emmons/writing/assignments/Hyde_Brainstorm.pdf`, last visited 2007-08-13, 2005.

IAF. *IAF Facilitator Certification Program*. International Association of Facilitators, 2004.

S.E. Jackson, K.E. May, and K. Whitney. Understanding the dynamics of diversity in decision-making teams. *Team effectiveness and decision making in organizations*, pages 204–261, 1995.

T. Jantunen. Methods for Requirements Elicitation. 2005.

C. Jones. *Software assessments, benchmarks, and best practices*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2000.

J.R. Katzenbach and D.K. Smith. *The wisdom of teams*. Harvard Business School Press, 1993.

Kelton Research. The Cyber Stress study. 2007.

G.L. Kolfschoten, R.O. Briggs, G.J. de Vreede, P.H.M. Jacobs, and J.H. Appelman. A conceptual foundation of the thinkLet concept for Collaboration Engineering. *International Journal of Human-Computer Studies*, 64(7):611–621, 2006.

P. Kroll and P. Kruchten. *The rational unified process made easy: a practitioner's guide to the RUP*. Addison-Wesley, 2003.

Dean Leffingwell. Calculating Your Return on Investement from More Effective Requirements Management. 1997.

N. Leveson and Clark S. Turner. An Investigation of the Therac-25 Accidents. *IEEE Computer*, 26(7):18–41, 1993.

Neil Maiden, Sharon Manning, Suzanne Robertson, and John Greenwood. Integrating Creativity Workshops into Structure Requirements Processes. 2004.

J. Martin. *Rapid application development*. Macmillan Publishing Co., Inc. Indianapolis, IN, USA, 1991.

K.D. Maxwell, L. Van Wassenhove, and S. Dutta. Software development productivity of European space, military, and industrial applications. *Software Engineering, IEEE Transactions on*, 22(10):706–718, 1996.

S. McConnell. *Rapid Development: Taming Wild Software Schedules*. Microsoft Press Redmond, WA, USA, 1996.

J.D. McKeen, T. Guimaraes, and J.C. Wetherbe. The relationship between user participation and user satisfaction: an investigation of four contingency factors. *MIS Quarterly*, 18(4):427–451, 1994.

S.M. Miranda and R.P. Bostrom. Meeting facilitation: process versus content interventions. *Journal of Management Information Systems*, 15(4): 89–114, 1999.

S. Mohammed and E. Ringseis. Cognitive Diversity and Consensus in Group Decision Making: The Role of Inputs, Processes, and Outcomes. *Organizational Behavior and Human Decision Processes*, 85(2):310–335, 2001.

B.A. Nijstad and W. Stroebe. How the Group Affects the Mind: A Cognitive Model of Idea Generation in Groups. *Personality and Social Psychology Review*, 10(3):186, 2006.

B. Nuseibeh and S. Easterbrook. Requirements engineering: a roadmap. *Proceedings of the conference on The future of Software engineering*, pages 35–46, 2000.

AF Obsorn. *Applied Imagination: Principles and Procedures of Creative Problem Solving'*. Charles Scribeners Sons, New York, 1953.

R. Purvis and V. Sambamurthy. An examination of designer and user perceptions of JAD and the traditional IS design methodology. *Information & Management*, 32(3):123–135, 1997.

QSM. *QSM SLIM-Estimate*. QSM, `http://www.qsm.com/slim_estimate.html`, last visited 2007-08-14, 2007.

B.A. Reinig, R.O. Briggs, and J.F. Nunamaker. On the Measurement of Ideation Quality. *Journal of Management Information Systems*, 23(4): 143–161, 2007.

S. Robertson and J. Robertson. *Mastering the requirements process*. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 1999.

Gary Rush. JAD - Its History and Evolution. *MGR Consulting Newsletter*, pages 2–5, July 2006.

J. Schalken, S. Brinkkemper, and H. Vliet. Assessing the Effects of Facilitated Workshops in Requirements Engineering. *Proceedings of the 8th Conference on Evaluation & Assessment in Software Engineering (EASE 2004), Stevenage, UK, IEE Press*, pages 135–144, 2004.

S. Schuman and IAF. *The IAF Handbook of Group Facilitation: Best Practices from the Leading Organization in Facilitation*. Jossey-Bass, 2005.

Standish Group. *The Chaos Report*. The Standish Group, 2000.

J. Stapleton. *DSDM: De Methode in de Praktijk, 2e Editie*. Addison-Wesley, 2005.

D. Straus and M. Doyle. *How to Make Meetings Work*. Jove, 1976.

W. Stroebe and M. Diehl. Why Groups are less Effective than their Members: On Productivity Losses in Idea-generating Groups. *European Review of Social Psychology*, 5(1):271–303, 1994.

R.I. Sutton and A. Hargadon. Brainstorming Groups in Context: Effectiveness in a Product Design Firm. *Administrative Science Quarterly*, 41(4), 1996.

D. Thomas and A. Hunt. The Pragmatic Programmer: From Journeyman to Master, 1999.

B.W. Unger and S. Walker. Improving team productivity in system software development. *Proceedings of the fifteenth annual SIGCPR conference*, pages 104–115, 1977.

Usability.gov. *Focus Groups - Usability Methods*. U.S. Department of Health & Human Services., `http://www.usability.gov/methods/focusgroup.html`, last visited 2007-08-21, 2007.

The University of Southern California USC. *CORADMO, The Constructive Rapid Application Development Model*. QSM, `http://sunset.usc.edu/research/CORADMO/coradmo_main.html`, last visited 2007-08-14, 1999.

Piet Verschuren and Hans Doorewaard. *Het ontwerpen van een onderzoek*. Lemma, Postbus 3320, 3502 GH Utrecht, Netherlands, 2000.

K.E. Wiegers. *Inspecting Requirements*. StickyMinds, `http://www.stickyminds.com/sitewide.asp?Function=WEEKLYCOLUMN&ObjectId=2697&ObjectType=ARTCOL`, last visited 2007-08-10, 2001.

K.E. Wiegers. *Software Requirements, Second Edition*. Microsoft Press, 2003.

L.A. Witt, M.C. Andrews, and K.M. Kacmar. The role of participation in decision-making in the organizational politics-job satisfaction relationship. *Human Relations*, 53(3):341, 2000.

J. Wood and D. Silver. *Joint application design: how to design quality systems in 40% less time*. John Wiley & Sons, Inc. New York, NY, USA, 1989.

J. Wood and D. Silver. *Joint application development, second edition*. John Wiley & Sons, Inc. New York, NY, USA, 1995.

E. Yourdon. *Death March*. Prentice Hall PTR, 2004.

P. Zave. Classification of research efforts in requirements engineering. *ACM Computing Surveys (CSUR)*, 29(4):315–321, 1997.

Appendix A

Annotated Bibliography

This chapter holds more information about the literature study. Section 1 presents the most relevant books and papers that have been studied. In Section 2 we discuss results of studying literature for finding reasons why workshops improve the quality of requirements. It provides background information to Chapter 7, section 1 about increased quality. Section 3 presents the background of workshops based on information found during the literature study. Section 4 presents the benefit statements as found in literature, and a categorisation we made to create the research questions.

A.1 Studied Literature

A.1.1 Studied Books

The most relevant workshop and requirement books studied during this research project:

Requirements by Collaboration (Gottesdiener, 2002)
This book is seen as the current day leading publication on using workshops for requirement engineering, and provides practical ideas about planning and leading workshop. It covers the six Ps in requirement workshops: Purpose, Participants, Principles, Products, Place, and Process.

Workshops: how to facilitate workshops in software engineering related environments (Hoogenboom et al., 2004)
A practical book explaining how to use workshops in software engineering projects. Besides discussing the art of facilitation and decision making, it explains how to lead workgroups. It gives a set of tools and practices

to solve problems and build teams. Has an extensive chapter on specific software engineering workshop tools.

Joint application development, second edition (Wood and Silver, 1995)
The often cited reference work about JAD. It introduces the JAD method, details the five phases, a look group psychology, and tools and techniques. While some content is slightly dated, and the book is no longer in print, it remains a must-read for facilitators.

Joint application design: the group session approach to system design (August, 1991). Another often cited book about JAD. Besides holding a treatise on JAD, it also includes information about JAD/Plan and JAD customisation, where these topics are missed in the above work. Also included are many templates that were used in the original JAD. Some information in this book is dated, though most information is still applicable and valuable for requirement workshop facilitators today.

Software Requirements (Wiegers, 2003)
This book was studied during the Master Software Engineering itself, and served as the reference for requirements during this project. It describes software requirements, requirement development, requirement management and how to implement requirement engineering.

Mastering the requirements process (Robertson and Robertson, 1999)
Another often cited book. It describes the popular volere requirement process and templates used in practice.

A.1.2 Studied Research Papers

The most relevant papers are:

Joint Application Design (JAD) in practice (Davidson, 1999)
Davidson investigated how workshops were used in practice and examined organisational constraints. Based on 34 interviews about 20 projects in three organisations, they found that JAD was most effective for small, clearly focused projects. Findings are that in only two out of 20 projects indicated that JAD helped define requirements faster. In only three projects was consensus on requirements achieved. Results suggest that the "JAD method is difficult to sustain in practice because assumptions underlying the method conflict with assumptions of the status quo development process in IS organisations, yet adapting the method limits its usefulness." (Davidson, 1999)

Assessing the Effects of Facilitated Workshops in Requirements Engineering (Schalken et al., 2004)
Schalken et al. evaluated the effectiveness of facilitated workshops and found that for larger projects (excess of 171 function points) facilitated workshops offer greater productivity than one-on-one interviews. This project measured results from 60 projects, though it was executed in one (large, bureaucratic) financial institution. The research was done at a time the institution was introducing workshops and DSDM, and moving away from Model/1 which used a traditional one-on-one interview technique to gather requirements.

An Examination of Designer and User Perceptions of JAD and the Traditional IS Design Methodology (Purvis and Sambamurthy, 1997)
Purvis et al. examined perceptions of workshops. Data was gathered from 94 pairs of designers and users and 57 project managers and projects using a questionnaire. It was found that "both users and designers agreed that the JAD methodology promoted greater and richer interactions (user participantion and influence, partnerschips, and good communication) among participants in the system design process. Designers perceived JAD to be superior in promoting effective consensus management and user acceptance. However, the user did not perceive any significant differences between the two methodologies on these two factors." Purvis and Sambamurthy found that while designers were less confident, and had less experience with the JAD methodology, still perceived JAD to be beneficial.

Higher Quality Requirements: Supporting Joint Application Development with the Nominal Group Technique Duggan and Thachenkary (2003)
Describes research and experiments done in workshop setting to decrease workgroup problems using the nominal group technique (NGT). The paper describes a set of 24 lab experiments involved generating and documenting requirements for a simulated case using workshops. Twelve experiments were facilitated with the use of the NGT, and twelve without. Though no higher levels of efficiency, it did find that the quality of resulting requirements were significantly increased when they used facilitation with NGT.

Improving Team Productivity in System Software Development Unger and Walker (1977)
This project researched facilitation of groups in an IT setting, groups of college students implementing software. Over a period of four years, the productivity of student groups with and without facilitated workgroup ses-

sions where measured. Group sizes ranged from 25 students in 1973 to eight or nine in the other experiments. Team-size was three students per team. Facilitated groups where found to be twice as productive. "The factor of two increase in individual programmer productivity in the 1973 project (a facilitated effort) is particularly surprising because three times as many programmers were involved."

Brainstorming Groups in Context: Effectiveness in a Product Design Firm Sutton and Hargadon (1996)
Based on analysis of 24 brainstorms using experienced brainstorm participants, meetings among other results "set, reinforce, and reflect organisation-wide values and norms, [... and] are an important (and efficient) means through which competition for status based on technical skill occurs between engineers." Also he indicates that clients were "not only often impressed with the concepts, prototypes, and finished products that resulted from the brainstorms, they were often impressed with the creativity display by [..] designers and the fun everybody had."

A.2 The Quality of Groupwork

Workshops proponents claim that the use of workshops in requirement engineering increases the quality of requirements. The use of workshops gets requirements faster, and by increasing the quality of requirements also saves time and money in the long run by preventing rework. To attain these benefits, the hypothesis must hold that workshops deliver a better quality requirement than alternatives like interviews. Strong evidence suggests that group work is not however productive. These findings are often based on assessing brainstorming.

Brainstorming is a method for creative problem solving, developed by Alex Osborn in 1939. Brainstorming was defined by Osborn as "a method by which a group tries to find a solution for a specific problem by amassing a list of ideas spontaneously contributed its members." (Hyde, 2005) Osborn noticed that the quality and quantity of ideas produced was much greater than those produced by individual employees, in fact a 2-1 productivity increase was found. Osborn published his method in 1953 in his book "Applied Imagination". (Obsorn, 1953)

Osborns rules for brainstorming sessions are: (Hyde, 2005)

- Judgment of ideas is not allowed (this comes later)
- Outlandish ideas are encouraged (these can be scaled back later)

- A large quantity of ideas is preferred (quantity leads to quality)
- Members should build on one anothers ideas (members should suggest idea improvement)

Findings from research since shows that face-to-face group brainstorm sessions are in fact ineffective. (see Sutton and Hargadon (1996) and Stroebe and Diehl (1994) for a list references) There are three reasons why productivity of groups is lower than that of individuals combining their ideas. First, evaluation apprehension, the apprehension of being judged. Productivity loss due to evaluation apprehension increases when authority figures are present during the brainstorm session. Second, free-riding or social loafing, where participants just do not participate as the group seems to do the work anyway. Finally, the strongest support for productivity loss is due to production blocking, waiting on each other before you get a turn to share your idea. (Diehl and Stroebe, 1987)

When people come together more than the previous three issues are found. To give an impression on the problems that can be experienced in groups, Duggan and Thachenkary (2003) list a set of nine commonly found problems, ranging from group biases, and dominance, to groups arriving at decisions that are contrary to the desires of its members.

Clearly many problems exist. To counteract the many possible problems, a workshop facilitator is used to guide the process to a successful completion. Miranda and Bostrom (1999) found that process facilitation improved groups' perceptions of their meeting process. Duggan and Thachenkary (2003) discovered that an improved process also improved requirements resulting from the process. Unger and Walker (1977) report on finding increased productivity with facilitated teams.

To illustrate the role of facilitation to solve often cited group brainstorm problems, take for example the biggest problem preventing group brainstorm productivity: production blocking, waiting on each other before you get a turn to give your new idea. (Diehl and Stroebe, 1987) If a facilitator gives all brainstorm participants a stack of sticky notes and a marker, and asks them to write down their ideas and stick them on a flip-chart, production blocking no longer plays a role. This method also creates a good way for cognitive stimulation, seeing the results of others triggers new ideas. (Nijstad and Stroebe, 2006)

To verify whether the facilitator can also get commitment if ideas are incorrect, the respondents of our survey were asked whether it happened regularly that good facilitators gets support and commitment for the *wrong* requirements. Of all participants 23% indicated that in fact facilitators could, of the other participants 42% was undecided and 35% disagreed. Respondents were also asked whether conflicting needs of participants caused problems

during the workshops that increased project risks. Only 12% indicated that it did.

Research results indicate that it is possible to improve the hard results of the group process, increase quality of results, and increase productivity through the use of group facilitation. It is evident though that a skilled facilitator is required to prevent problems. Also, Sutton and Hargadon (1996); Mohammed and Ringseis (2001); Nijstad and Stroebe (2006) report that soft results are significantly improved with group sessions. And it is these results that influence whether or not a user is involved with the project, and committed with its results.

A.3 History of Workshops

The use of collaboration to create system requirements can be traced back to 1977, IBM's Milwaukee, Wisconsin office, and to Chuck Morris. The developers were having trouble deciding how to implement a system called COPICS. This was an early Manufacturing Resource Planning (MRP) system. To gather requirements for screen design and process design from people who had little or no understanding of computers, Chuck suggested sitting developers down with representatives of the user community to talk about requirements. (Wood and Silver, 1995; Rush, 2006)

Many of the roles for this process were based on a book written a year before: 'How to Make Meetings Work' by Straus and Doyle (1976). This approach, now called Joint Application Development (JAD), was also loosely derived from another IBM methodology, Business Systems Planning. (BSP) (Rush, 2006; Carmel et al., 1993)

In 1979, Tony Crawford of IBM Canada helped Chuck Morris formalise the process. Tony developed the JAD-Plan agenda, a workshop to help planning for the JAD workshops, and implemented JAD at the IBM Canada office in Toronto. (Rush, 2006)

Chuck moved to IBM's Raleigh office in Research Triangle Park, North Carolina, and the first JAD meetings were held at IBM's Raleigh offices in design of a distribution system called Distribution Center Operations Workshop. "This project used the same basic concepts used today: user participation meetings, magnetic visual display, and documentation of the meeting in workshop reports." (Carmel et al., 1993)

In the middle 1980s and early 1990s, Rapid Application Development (RAD) became popular, based on the book "Rapid Application Development", by Martin (1991) "JAD became a favoured way to define user requirements quickly. As object-oriented development and the Internet became widely

used, JAD reemerged as a way to accelerate delivery of requirements." Gottesdiener (2002)

In 1994 the workshop technique was adopted the DSDM Consortium that was formed in the United Kingdom. The method was called the Dynamic Systems Development Method. (DSDM) The first version of the manual was published in 1995. (Stapleton, 2005) "Prototyping and JAD continue to be used today, especially when requirements are poorly understood." (Glass, 2003)

A.4 Workshop Benefits

Studied literature states many benefits of using workshops. To create the right research questions, benefit statements were collected, and a categorisation of these benefit statements was made. The three categories we found were stated as research questions for this research project.

This categorisation is based on benefits reported by Wood and Silver (1995); Gottesdiener (2002); DSDM Consortium (2003); Carmel et al. (1993).

- It involves the user, which:
  - Improves relations between business domain and IT.
    - It closes the we-versus-they gap, by working together IT and business representatives.
    - It establishes mutual understanding.
    - It builds relationships.
  - Increases stakeholder commitment and buy-in.
    - Greater user buy-in as stakeholders are directly involved in system design.
    - The user designed the system; therefore it is their system.
- Accelerate system design.
  - It reduces the time to gather requirements, increasing team productivity.
  - Because everybody is present, it shortens lines of communication.
  - Rapid decision making. The group is focused on the objectives so that information gathering and review cycles are performed at greater speed.
  - Workshops consolidate months of design meetings, follow-up meetings, review meetings and telephone-call meetings 'to clear this up' into one structured workshop.

  - Decreased development costs.

- Improve the quality of system design.
  - Improved accuracy of requirements
  - It brings together stakeholders, and sit eyeball-to-eyeball to discuss the project.
  - Stakeholders work together to present viewpoints, answer questions and create mutually agreeable solutions.
  - Enables team members to obtain an overall view of the product requirement in large or complex projects

Appendix B

Interview Results

B.1 Organisations and Workshop Adoption

Organisations that were interviewed where chosen because they are using workshops. The various organisations of which representatives were interviewed showed different ways of using workshops to gather requirements.

Four of the five representatives of internal IT departments indicated that workshops were used as a standard technique for gathering requirements. Only the smallest internal IT department did not use workshops as a standard technique, they used workshops on an as needed base driven by individual experience with workshops. The larger IT departments used as default part of the requirement process.

The large IT automation organisations, that were contacted, all were international organisations. All of these organisations used workshops as a default technique to gather requirements. The interviews indicated that workshops were most often used during early requirement engineering phases, like setting goals and project scope. Though workshops were used throughout the requirement phase, and in some situations even throughout product development.

Representatives were interviewed of two smaller IT automation organisations. Both organisations used workshop techniques as a tool in the requirement process. One used workshops more extensively as a way for gathering requirements, prototyping and reviewing prototypes. The other used workshop techniques less often, for more specific tasks like prioritisation of requirements with several stakeholders. Both organisations used experience of the requirement engineer whether or not activities used workshop techniques.

Of the requirement specialists, both indicated using workshops extensively to gather requirements. These specialists did not implement the products,

only engineering requirements. Processes resembled those of the large IT organisations.

One large organisation had a different business unit and different workshop leaders, for the different stages of software development. More often, the business analyst lead the first few higher level workshops, and project managers took over from there. Most organisations used a core team that participated in the workshops throughout the project. This often consisted of a project manager in IT domain, project manager (or visionary) of business domain, business or IT architect, requirement engineer, business or information analyst. Mid-level management, subject matter experts, and end-users were invited as required. Individuals could fulfill more roles than one.

To conclude, all of the larger IT organisations (and larger in-house IT departments) contacted indicated using workshop techniques as their default technique. Also, the requirement specialists used comparable processes and techniques to gather requirements. Some interviewees of these organisations even stated that "no requirements without workshops." Based on interviews we conclude that of the contacted organisations, those with a more developed requirement process used workshops extensively, where others used workshop techniques more based on individual experience and as needed.

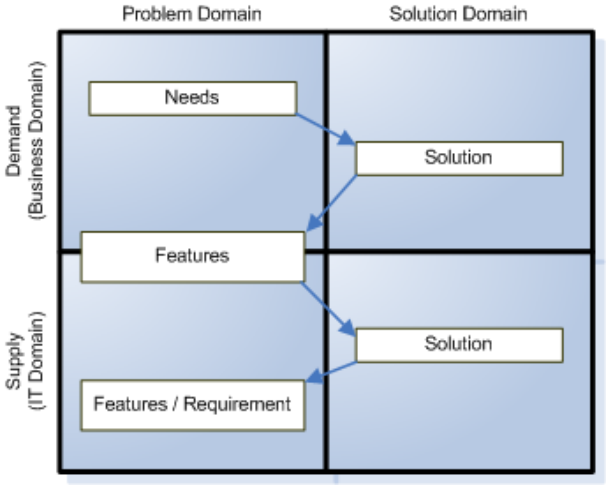
B.2 The Types of Requirement Workshops

The single type JAD workshop as described by Wood and Silver (1995) was not found in practice. Of all organisations only one held something resembling the big multiday JAD workshops, but not specifically for requirements. Other organisations all used a series of shorter workshops (often half a day, at most a day) to gather the requirements.

The JAD/Plan and JAD/Design division as described by August (1991) was seen in practice. Though the two level distinction of Plan/Design was seen in larger organisations as a three level activity. First the planning workshops to set goals, then translate them to scenarios and processes, finally work out details and prototype design. This model is displayed in Figure 8.2 on page 58.

The model in Figure 8.2 is similar in a way to the top-down navigation strategy as described by Gottesdiener (2002). This model maps the process of decision making and detail specification to different levels of the organisation. The middle layer of middle management, where high-level needs and goals are translated to applicable solutions is more useful than the dual-layers approach by August (1991) for larger organisation and more difficult projects.

Figure B.1: Needs/Solution Square Model



This model raises the question on how decision making is done with the use of workshops. On the right side of the model it shows the presentations of the results to the higher levels. For multi-million euro projects decisions will not be made by the normal every day worker. A feedback loop is needed to get the go-signal of higher management.

Feedback can either be done as shown, using presentations, or using the project method of the project like Prince2. In this case, big decisions are escalated to higher levels of management for the actual decision making. Also, feedback of events will have to be shared with management as well. This can be done using the workshop report. Though a big list of requirement statements will not be very helpful as communication mechanism to management.

B.2.1 Activities

Wim Dijkgraaf and Mike van Spall developed a model that shows the activities of Plan and Design. Figure B.1 shows the model, adapted to suit the following explanation. The original was recently published in Dijkgraaf and Van Spall (2007). Two levels are shown: one of the Demand (business domain) which experiences a problem or need, and one of supply (often IT domain). It also acknowledges the problem and solution domain.

This model shows that requirements definition is an activity of problem solving in the business domain. All too often the problem and business domain is ignored, and requirement engineering focuses on solutions and features. Several interviewees told that IT is often only called in to create a solution, not solve the problem. While interviewees found that benefits could

be had of IT would have been involved during the creation of the solution ideas. It was seen as difficult to persuade the business domain stakeholders to go back to look at business goals, business area processes, and evaluate different types of solutions.

This model is also interesting because it clearly shows iterations towards the final solution statement. In this diagram the first iteration consists purely in the demand domain. It looks at vision, goals and needs, not at systems and solutions. Once the needs are clearly defined, a solution idea is defined using creative brainstorming and some form of decision making. This is done using workshops with the, at that point, relevant stakeholders. Once a solution idea is created, then the required features are found for the created idea. Once the features are found, a details solution is created to fit the required features.

Two different style of workshops are displayed: creative solution finding shown as arrows to the right, and solution defining workshops shown as arrow pointers to the left. This model shows that this is a iterative, and thus repetitive, activity.

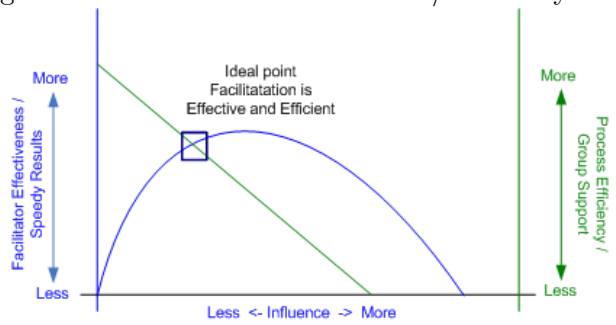
B.3 Collaboration Engineering

The collaboration processes and tools, and the style of facilitation by the facilitator, influences both the hard results (decision, plans, requirements made during the session) and soft results (how participants feel about the hard results) to a large degree. Decisions can be made fast by overruling opinions of individual, or can be made in group consensus. The group consensus style increases decision satisfaction (Mohammed and Ringseis, 2001) but does often require more time. Figure B.2 shows the influence of a facilitator, and the efficiency and effectiveness in a model. Also facilitators that intervene on content (for example taking decision, or steering a group to a predefined outcome) are seen as giving a negative impact in the group process, decreasing satisfaction with the process and the results. (Miranda and Bostrom, 1999)

Results of workshop depend at least on the procedure (what happens) and the process (how it happens). Whether or not facilitators are aware of this, with their plans, actions and decisions they are actively engineering the collaboration. One facilitation is different from the other, even though on paper they seem the same.

To improve the understanding of facilitation and group processes, the research field of collaboration engineering is concerned with designing processes and making them transferable (and repeatable) over different groups and facilitators. De Vreede and Briggs (de Vreede and Briggs, 2005) have de-

Figure B.2: Facilitator Effectiveness/Efficiency Model



veloped a method of designing a collaboration which acknowledges the three levels: freely interpreted to mean the What, the How and the Activity.

As part of their research they have created a pattern language for collaboration called thinkLets. Comparable to the collaboration patterns as defined by Gottesdiener (2002) and building blocks by Hoogenboom et al. (2004) thinkLets define detailed processes used for collaboration and facilitation. "A thinkLet is a named, packaged facilitation technique that creates a predictable, repeatable pattern of collaboration among people working towards a goal". (Briggs et al., 2001)

Though their research is not oriented towards requirement engineering, some effort has been taken to incorporated apply collaboration engineering to requirements. (Alaa and Appelman, 2006; Grunbacher and Briggs, 2001) The field of collaboration engineering research is very applicable to requirement workshops.

B.4 Accelerated Design

Another factor was found the influences whether or not workshops reduce the duration of requirement engineering: the intensity and frequency with which the workshops are held. In the early 1990s MG Taylor Corporation in combination with Ernst & Young, AT Kearney and CSC Index developed the Accelerated Solution approach. These workshops have been engineered to be a highly intensive, focused and facilitated series of collaborations in a special environment suited to collaboration, lasting for two or three days.

The intensive collaboration is facilitated by a team of professional facilitators. These include the actual facilitators, support facilitators, psychologists, software developers, architects, graphical designers, and cartoonists. Together they design and execute the session which includes everything from envisioning goals, solving problems, designing solutions and iterating over previously these solutions. This is mixed with eating and drinking, dancing,

training, presentations, meditation, and everything else that is necessary for maximum efficiency and effectiveness.

During this project we found that workshops were often held over multiple weeks, with between one and three workshops per week, lasting only a few hours. The accelerated solution workshops compacts these week long processes into a few days. This system holds the possibility to significantly shorten the duration of the requirements phase. An exploratory experiment, to see what happens when the requirement phase is compressed using these techniques, can be found on page 103.

Appendix C

Requirement Workshop Questionnaire

The survey was held using a web-based survey tool, with the option enabled to skip questions, but not the statements. Also, all questions in a category were shown at once. Next and previous category buttons were present, together with an option to save the survey and continue later. An indicator at the top showed the participants progress through the questions.

In total 53 questions were asked in seven categories. Eight questions about workshop use, nine questions about the dimensions of workshops, five questions about the preparation, five questions about the content, five questions follow-up, six questions about quality, and finally seven questions about personal experiences.

C.1 Survey Questions

The requirement workshop questionnaire is shown in the following sections. Unless stated otherwise, questions are open questions; a text area was shown where respondents could enter answers.

C.1.1 Workshop Use

U.1: *For which requirement activities do you use workshops?*
A multiple answer question, options are: Creating vision, Creating business models, Identifying stakeholders / user classes, Establish business goals / business case, Develop use cases, Identify and solve problems, Specifying use cases, Specifying business rules, Finding new requirements, Understanding priorities, Decision taking, Business process (re-) design, Determine scope, Creating prototype, Validation.

U.2: *Which methods for finding new requirements do you often use besides workshops?*

U.3: *Which types of projects do you usually do?*
Multiple answer question, options are:

- Creation of new systems
- Modification of existing systems
- Software product evaluation / selection
- Software products configuration

U.4: *Which types of software do you usually develop?*
Multiple answer, options:

- Management Information Systems / Administrative Software (Payroll, Accounting)
- Embedded, Real-time or Control Software (Monitoring, switching systems)
- User Tools Software (Report generators, Wordprocessing, Spreadsheet)
- Developer Tools Software (Product generators, code generators, testing software)
- Utility Software (backup, installation, conversion utilities)
- Systems Software (operating systems, printer drivers, protocol converters)

U.5: *Please rank the following reasons you use workshops. Start with the most important reason.*
Items to rank are: Determine goals, Improve problem solving, Reduce time to market, Decrease number of missing requirements, Decrease number of requirement changes, Safely surface conflicting needs, Create a common view, Improve stakeholder commitment, Improve user communication.

U.6: *Which downsides do you experience from the use of workshops?*

U.7: *What does your standard requirement gathering process look like?*

U.8: *Statement: Using workshops reduces the requirement effort*
Multiple choice: Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree.

C.1.2 Dimensions

D.1: **How many participants attend a workshops?**

D.2: *How many facilitators lead the workshops?*

D.3: *How many requirement workshops are held in one project?*

D.4: *How many hours does a requirement workshop take?*

D.5: *How many requirements are created during one workshop?*

D.6: *How many use cases / scenarios / process descriptions are created during one workshop?*

D.7: *What percentage of total project time is used for the requirements phase?*

D.8: *How much time do you spend on workshops (including workshop preparation) compared to other requirement tasks?*
Multiple choice: 100% Workshops 0% Other, 80%-20%, 60-40, 50-50, 40-60, 20-80.

D.9: *Statement: Interviews are better in finding and understanding new requirements.*
Multiple choice: Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree

C.1.3 Preparation

P.1 *What tasks do you do before the workshop?*

P.2 *What tasks do the participants have to do before the workshop?*

P.3 *What is the quality of the pre-work tasks of the participants?*
Multiple choice: Excellent, Very good, Adequate, Weak, Poor

P.4 *How many hours of preparation work would be required for a 1 day (8 hour) workshop?*

P.5 *Statement: Getting the right participants is often impossible.*
Multiple choice: Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree

C.1.4 Content

C.1: *Which models / techniques do you use in the requirement workshop?*
Multiple answer, options are: Actor table, Persona's, Prototype, User interface navigation diagram, Dialog Hierarchies, Relationship map, Glossary, Data Model (Entity Relationship Diagram (ERD) or class model), Event / response table, State Diagrams, Sequence Diagrams, Architecture Diagrams, Business Process diagram, Business rules, Context diagram, Decision tables / trees, Process map, Domain model, Scenarios / Stories, Activity diagram, Data Flow Diagram, Use cases.

C.2: *What other models do you use?*

C.3: *Which types of participants do you invite to the workshop?*

C.4: *What portion of the requirements created with workshops are vague and untestable*
Multiple choice: 0-5%, 5-10%, 10-20%, 20-30%, 30-40%, 40% or more

C.5: *How many requirements are changed (added, removed, modified) after the workshops?*
Multiple choice: 0-5%, 5-10%, 10-20%, 20-30%, 30-40%, 40% or more

C.6: *Statement: Participants can't judge the correctness of information systems (IT) models that were created during the workshops.*
Multiple choice: Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree

C.1.5 Follow-up

F.1: *What are the main deliverables from the workshop?*

F.2: *Which tasks do you do after the workshop?*

F.3: *What follow-up activities are required from participants?*

F.4: *How many hours of follow-up work would be required for a 1 day (8 hour) workshop?*

F.5: *Statement: It happens regularly that real commitment is missing.*
Multiple choice: Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree

C.1.6 Quality

Q.1: *When are workshops NOT the best method for finding requirements?*

Q.2: *If you would not use workshops, how would this impact quality of requirements?*

Q.3: *How do you evaluate the requirement workshops?* Multiple answer: Participant satisfaction, Workshop debriefing, Measure requirement stability / changes, Measure effort, Use of models / diagrams in software design and construction, Shared vision, Level of commitment, Number of deliverables (e.g. number of business rules, use cases, requirements), Results of user (verification/validation) tests.

Q.4: *Other evaluation activities you take*

Q.5: *Statement: It regularly happens that good facilitators get support and commitment for the WRONG requirements.*
Multiple choice: Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree.

Q.6: *Statement: Putting stakeholders with conflicting needs together in a workshop will cause problems that increase project risks.*
Multiple choice: Strongly Agree, Agree, Neither Agree nor Disagree, Disagree, Strongly Disagree.

C.1.7 Personal

E.1: *What do you do different now from what you did in the past?*

E.2: *How many years of experience do you have facilitating workshops?*

E.3: *How many workshops did you hold in the last year?*

E.4: *How suitable would you say the use of workshops is for finding new requirements?*
Multiple choice: Excellent, Very good, Adequate, Weak, Poor

E.5: *What training did you have about workshop facilitation?*

E.6: *Which reference material do you use? Books, training material, and so on. Please state books and other material you find important.*

E.7: *What would you like to know about requirements engineering or workshops?*

Appendix D

Gathering Empirical Data

Part of this research project was trying to measure to which extend workshops aided in decreasing requirement volatility and shortening both the requirements phase of a project, and the entire project due to the increased quality of requirements. For this, a template was created. Table D.1, D.2 and D.3 show the (partial) gathered data from a project using requirement workshops that was analysed during this research project.

Table D.1: Projects Data

Project					
		Company	IT development company		
		Duration	85 days		
		Project type	Modification / Extension		
		Software Type	Information Systems / Payment		
		Method	Incremental		
		Methodology	Based on DSDM		
		Branch	Banking		
Dimensions					
	Project:	Cost	€ 600.000,00		
		Duration	17 weeks		
		Total dev effort			
	System:	Involved depts	2		
		# Users (est.)	5500		
		Lines Of Code	243100 (excl frontend JSP code, incl database code & JUnit tests)		
		Language	Java		
		FP	600		
Reqs	**Reqs**	Tot # Reqs	-		
		Average Wordcount	-		
	NFRS	Tot # NFRs	-		
		AVG WC	-		
	Use cases	Tot # Ucs	9		
		AVG WC	680		
	Req Validation / Acceptation				
		Effort	?		
		Duration	1 week		
		Activities	Email reqdoc, process feedback		
Delays					
		Effort	0		
		Duration	0		
Phases					
	Introduction to project				
		Duration	4 days		
		Effort	14 hours		
		Activities	Read existing documentation / plans		
	Req Phase	# req ws	4 (3 increments + 1 priorities)		
		Duration/ws avg	6 hours		
		Fac. Effort	155 hours		
		Duration	6 weeks		
		Activities	Verify usecases based on vision		
	Development				
		Effort			
		Duration			
	Validation & Acceptation				
		Effort			
		Duration	1 week		
		Activities	1 day user acceptation test with stakeholders on local server, 6 participants of which 4 from customer		
			3 days production acceptation test by hosting provider		

Table D.2: Workshop Data

Workshop			1. First Increment Workshop	2. Second Increment Workshop	3. Third Increment Workshop
Preparation					
	Total duration		9 days		
	Total effort		61		
	Interviews				
		#	2, with visionary of bank	0	0
		Effort	17 hour	-	-
		Duration	2 days	-	-
		Activities	Prepare, discuss vision and plans, create workshop plan.		-
	Pre Session Meetings				
		#	0	0	0
		Effort / Duration	-	-	-
		Activities	-	-	-
	Workshop prep activities				
		Activities	Create init req doc, ws presentation	Create init req doc, ws ppt	Create init req doc, create ws ppt
		Effort	20 hour, 28 hour	15 hour, 15 hour	15 hours, 8 hours
		Duration	6 days	4 days	4 days
Workshop					
		Type	Walkthrough/Verification	Walkthrough/Verification	Walkthrough/Verification
		Duration	6 hours	6 hours	5 hours
		Activities	View/discuss created scenario's based on vi	View/discuss created scenario's based on vision	View/discuss created scenario's based on vision
		Deliverables	Soft:Commitment Hard: req doc	Soft:Commitment Hard: req doc	Soft:Commitment Hard: req doc
		Models used	Scenario's and Screenshot prototypes.	Scenario's and Screenshot prototypes.	Scenario's and Screenshot prototypes.
	Facilitators	#	1	1	1
	Scribe(s)		Yes, a requirements engineer	Yes, req engineer	Yes, req eng.
	Participants	#	10	10	10
			Bank Business Analist, Bank Project Leader, Bank IT Architect, bank: 2 representative of user groups, bank: 2 functional maintainers.		
			IT Facilitator, IT SW Architect, IT Scribe/ReqEng,		
	New Requirements				
		#	-	-	-
		Average Wordcount	-	-	-
	New Scenarios/Use Cases				
		#	2	6	1
		Average Wordcount	800	410	2000
	Changed Reqs				
		# changed	-	-	-
		# removed	-	-	-
	Changed Scenarios/UC				
		# changed	-	-	-
		# removed	-	-	-
	Evaluation				
		Type	no workshop evaluation	no workshop evaluation	no workshop evaluation
		Results	-	-	-
Follow-up					
	Final Document				
		Activities	Process fdbk in existing doc.	Process existing req doc	Process existing req doc
		Duration	4 hour	4 hour	4 hour
		Effort	1 day	1 day	1 day
	Document Review				
		# of reviews	1	1	1
		Activities	Per email, feedback.	Per email, feedback	Per email, feedback
		Duration	1 week	1 week	1 week
		Effort	?	?	?
	Other / Misc				
		Duration			
		Effort			
		Activities			

Table D.3: Volatility Data

Requirements Volatility			1. First Increment Workshop	2. Second Increment Workshop	3. Third Increment Workshop
Dimensions					
	during req phase				
		# corrections	not available	not available	not available
		# additions	-	-	-
		# removals	-	-	-
	req eval / valid / accep				
		# corrections	6	8	3
		# additions	4	4	8
		# removals	-	0	0
	on baseline during dev.				
		# corrections	not available	not available	not available
		# additions	-	-	-
		# removals	-	-	-
RFCs/Bugs in use (first year)			not available, system in development		
	# RFCs				
	# bugs/errors				
	Type				
	Cause				
	Scope				
	Out of scope				
	In Scope				
	Defined as req, not analyzed				
	Def as req, not implement				
	Def as req, impl incorrect				
Notes			Analysis of changes in worddocument, including answers to open issues and misc additions		

Appendix E

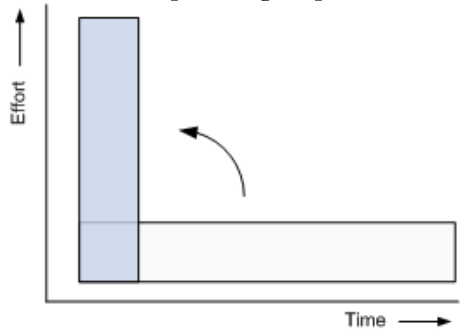
The Shortened Inception Experiment

During interviews one clear approach was found that could shorten the duration of requirement gathering. The intensity and frequency with which workshops are held determine for a large extend whether or not time can be saved. Based on interviews and the survey, we found that all but one organisations contacted used a spread-out model for planning workshops. Workshops are usually held two or three times a week, often lasting 3 or 4 hours.

In the early 1990s the Accelerated Solution approach was created. This method includes a highly engineered, intensive, focused and facilitated series of collaborations in a special work environment. The accelerated solution workshops are not used specifically for software development, but they can be used in software development projects as well.

A series of weeks or even months of workshops are compressed to two or three days of continuous collaborative sessions. Often 70 and more workshops are

Figure E.1: Compressing requirement effort.



held over these days, with over 50 participants. As there is no time to wait for ideas to develop, a collaboration process is created to speed this up. A highly experienced team of facilitators, IT experts, business analysts, psychologists and even cartoonists come together to create a process and an environment to facilitate accelerated solution design. (Capgemini, 2007)

Evidence of logical, commercial and anecdotal nature suggests that this method can save on the duration of the requirement phase. As we hypothesize that shortening the requirement phase saves money, an experiment with a project estimation tool was executed.

E.1 The Experiment

A project of 500 function points was estimated using a leading project estimation tool. QSM SLIM-Estimate is based on experiences from well over 6000 projects. (QSM, 2007) First, a normal scenario was estimated with the optimal balance between duration/effort. Results of this scenario are shown in Table E.1.

Table E.1: Project estimation numbers.

	Normal	**Shortened**
Life Duration (Months)	6,1	6,1
Life Effort (PHR)	3.184	2.415
Life Average Staff (ppl/month)	3,03	2,18
Inception Duration (Months)	0,9	*0,45*
Inception Effort (PHR)	297	297
Inception Peak Staff (people)	2	3,80
Elaboration Duration (Months)	1,3	1,4
Elaboration Effort (PHR)	530	389
Elaboration Peak Staff (people)	2,3	1,6
Construction Duration (Months)	3,2	3,5
Construction Effort (PHR)	2.121	1.554
Construction Peak Staff (people)	3,8	2,6
Transition Duration (Months)	0,7	0,8
Transition Effort (PHR)	236	175
Transition Peak Staff (people)	2,1	1,4
Productivity (total hr/fp)	6,4	4,8
Cost	€318.400	€241.500

As experiment, the duration of the requirement phase (the Inception phase) was halved. In this experiment we hypothesized that effort would not de-

Figure E.2: Reduced Inception duration project estimation.

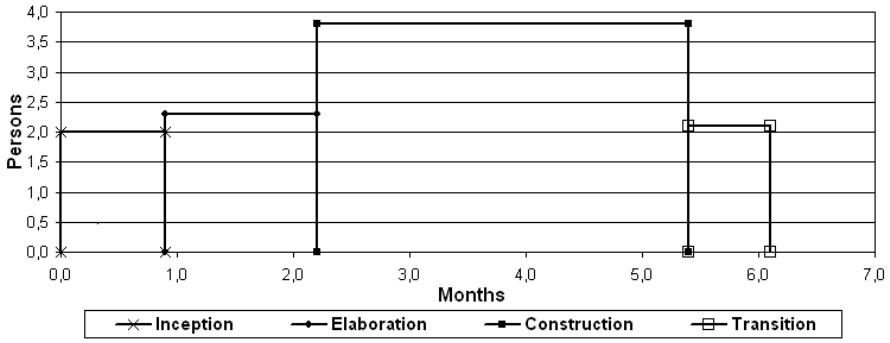
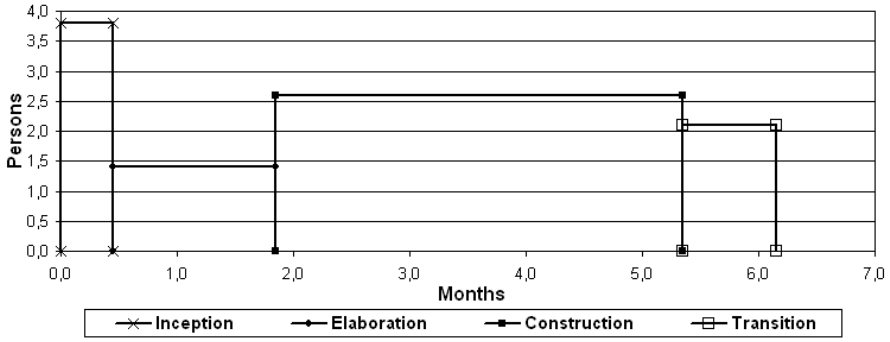


Figure E.3: Industry standard project estimation.



crease as the workshops required increased levels of facilitation. A fixed time scenario was simulated, no changes where made to the final deadline.

The results of the reduced Inception scenario are displayed in Table E.1. Results include a projected 25% decrease in hours of effort per function point. Productivity improved, from 6.4 hours per function point to 4.8 hours per function point. Furthermore a 30% decrease in required peak staff during construction phase was found.

The benefits resulted in an overall cost saving of €76.900, which is a saving of 24%. It can be concluded that productivity can be improved, and money can be saved using workshops.

E.2 Verification and Validation

These results were verified with another estimation tool, the Constructive Rapid Application Model (CORADMO). CORADMO is an adaptation of the Constructive Cost Model II (Cocomo) to Rapid Application Development projects. (USC, 1999)

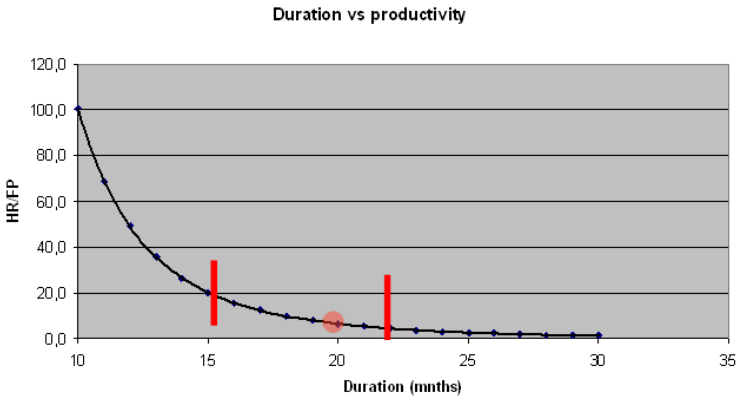
Cost and effort savings could not be duplicated with CORADMO, but the decreased staff during construction could. The last update to CORADMO was in 1999, which means that its data is 8 years old. To explain differences, we hypothesize that CORADMO uses a straight duration-productivity line. Shortening one phase and extending another, does not change total development effort.

In practice, this 1-1 trade-off between duration and productivity does not hold. To deliver the same product in less time means more developers are required. Brooks (1995) reports that man-months are mythical, because larger teams are not equally more productive. Maxwell et al. (1996), based on analysing 99 software development projects, found proof that productivity decreases with larger teams. They found that if team size is increased by 10%, productivity will decrease by 5%. They hypothesize that this is "due to the coordination and communication problems that occur as more people work on a project."

Figure E.4 shows the effects on productivity that is to be expected when changing the duration of a project. The numbers on the axis depend on diverse project characteristics. Shortening or extending the duration beyond the vertical bars results in either significantly decreased chances of project success or very marginal improvements.

The big win in shortening the Inception phase and extending the Construction phase is in this curve. You move over the curved line to the right, requiring lower productivity, and needing a smaller development team, but without extending the total project's duration. As productivity increases,

Figure E.4: The Duration-Productivity model.



the effort that is saved by using a smaller development team is above that of the increased duration of construction.

E.3 Conclusions

This experiment halved the Inception phase on the basis that time could be saved in gathering requirements. The Inception phase of a Rational Unified Process (RUP) project requires more activities than just gathering requirements. The Inception phase also includes "acquiring or implementing key elements of the architecture, or different suggested architecture, to better understand the risks and options you might have." (Kroll and Kruchten, 2003)

The results of cost saving were not duplicated with the CORADMO tool. This difference was accounted for by CORADMO using a fixed duration-productivity ratio. Also, this experiment is based on simulation of a project, and did not measure any practical aspects.

As the estimation tool is one of the market leading products, is based on data from over 6000 projects, and was operated by an experienced professional during this experiment, the results of this experiment are indicative of what happens in reality. Results suggest that the effects of compressing the Inception phase can save money and increase productivity.