Requirements Eng (2012) 17:157–170
DOI 10.1007/s00766-011-0131-2

ORIGINAL ARTICLE

Comparing task practicing and prototype fidelities when applying scenario acting to elicit requirements

Gyda Atladottir · Ebba Thora Hvannberg · Sigrun Gunnarsdottir

Received: 29 December 2010 / Accepted: 27 July 2011 / Published online: 13 August 2011

Abstract Identifying accurate user requirements early in the design cycle is of the utmost importance in system development. The purpose of this study of requirements elicitation was to compare the results of involving the user early in the design cycle using a low-fidelity prototype with the results of involving the user after a high-fidelity prototype was available. Three groups of potential users applied the method of Scenario Acting. Participants in the first group were given a working prototype of a human capital development system. The participants of the second group were given a detailed description of proposed features of the system and were told to practice on a paper prototype or with current methods, such as an Internet browser. These groups then practiced the tasks for some time before participating in the Scenario Acting. The third group received a brief description of the objectives of the system and did not practice the tasks. The results of the study showed that the use of the high-fidelity prototype was not helpful for eliciting requirements when working with users. However, the second group, taking time to practice the tasks given a low-fidelity paper prototype outperformed the others. Furthermore, the analysis of the Scenario Acting sessions revealed that two sessions were better than one, especially for participants of the group working with a low-fidelity prototype. An analysis of the topic of requirements showed that there was no difference between the groups on the domain tasks (here, human capital development), but the group practicing on the high-fidelity prototype commented more on its ease of use and usefulness than the other two. By comparison, the group practicing on low-fidelity prototype had more comments on the practice of the work and output of the tasks.

Keywords User requirements · Scenario · Scenario acting · Role playing · Requirements elicitation

1 Introduction

Project managers cite badly defined or excessive requirements as causes of software project failure along with factors such as poor management, lack of budget and lack of necessary technical skills [1–4]. Although badly defined requirements are not the only cause of failure, it is one of the topmost causes and, therefore, it is of the utmost importance for software quality that the requirements of a system be clearly identified. Starting with requirements that represent and closely address a user's needs is far less costly than having to make changes after actual programming is well underway. Boehm and Papaccio [5] estimated that it costs 50–200 times more to fix or rework software in the later stages than in the earlier phases of the life cycle.

Different terms are used for the act of determining which features should be implemented, including requirements gathering, requirements elicitation and requirements analysis. Preece et al. [6] chose the term 'establishing requirements' to indicate that the requirements are the result of gathering and interpreting data and that the requirements represent the correct understanding of the needs of the user:

> "Requirements determination is about social, communication, and managerial skills. This is the least technical phase of system development, but if not

G. Atladottir · E. T. Hvannberg (✉)
University of Iceland, Dunhaga 5, 107 Reykjavik, Iceland
e-mail: ebba@hi.is

S. Gunnarsdottir
Maritech, Reykjavik, Iceland

> done thoroughly, the consequences are more serious than in other phases. The downstream costs of not capturing, omitting or misinterpreting requirements may prove unsustainable later in the process" [6].

Cheng and Atlee [7] summarised the state of the art of requirements engineering for each of the five requirements tasks, elicitation, modelling, requirements analysis, validation and verification and requirements management. According to their classification, requirements elicitation touches upon different techniques, including (1) identifying stakeholders, (2) applying analogical techniques like metaphors and personas, (3) techniques that analyse stakeholders' requirements in a particular context, (4) techniques for inventing requirements, such as brainstorming, and (5) feedback techniques to elicit feedback on early representations of the proposed system.

The work reported in this paper aimed to research techniques in the last three categories of requirements elicitation. By providing users with relevant tasks to practice before a requirements elicitation session, the objective was to determine whether familiarising users with a particular context would assist the requirements elicitation process. Second, a technique called Scenario Acting, in which users act out requirements scenarios to invent requirements, was used. Finally, this research examined the helpfulness of providing users with a prototype to stimulate feedback on the proposed system.

Involving users has long been recognised as an intrinsic part of the requirements phase [8]. Kujala [9] states, '[S]ome evidence exists to suggest that taking users as a primary information source is an effective means of requirements capture'. There are many challenges associated with involving users in the requirements analysis, but their involvement has been shown to affect the final product in a positive way, to contribute to user satisfaction and to increase the accuracy of the requirements. Researchers have invented various methods to involve potential users in eliciting requirements, such as observing users at work and/or carrying out interviews with them related to the tasks they are performing. A third method, role playing, joins developers with a group of potential users. The objective is to create group dynamics that are expected to increase creativity and elicit better ideas from users. Role playing has been used as a technique since the 1980s [10] and has been researched in several different studies [11–13]. For the designer to understand users' implicit and non-verbal needs, she may ask them to role play specific use-cases of the product. That is, the designer asks users to create and act out a scenario that relates to a task or tasks that may be implemented and supported by the system being developed.

Although the role playing technique has been the subject of several papers, Svanæs and Seland [12] did not identify any detailed guidelines on how to apply this technique. They filled this gap by providing lessons they learned from conducting six workshops. A similar technique, termed Scenario Building, weaves together three aspects (users, tasks and situations and the product), and the product may be a loosely defined concept or a prototype [14].

Prototypes are frequently used in software development, either as low-fidelity prototypes that are developed early for requirements elicitation or validation or as high-fidelity prototypes for validation. The use of prototypes for user-interface evaluation has been studied [15, 16], but as far as we can see, few comparative studies including prototypes have been carried out for general requirements engineering [17–19].

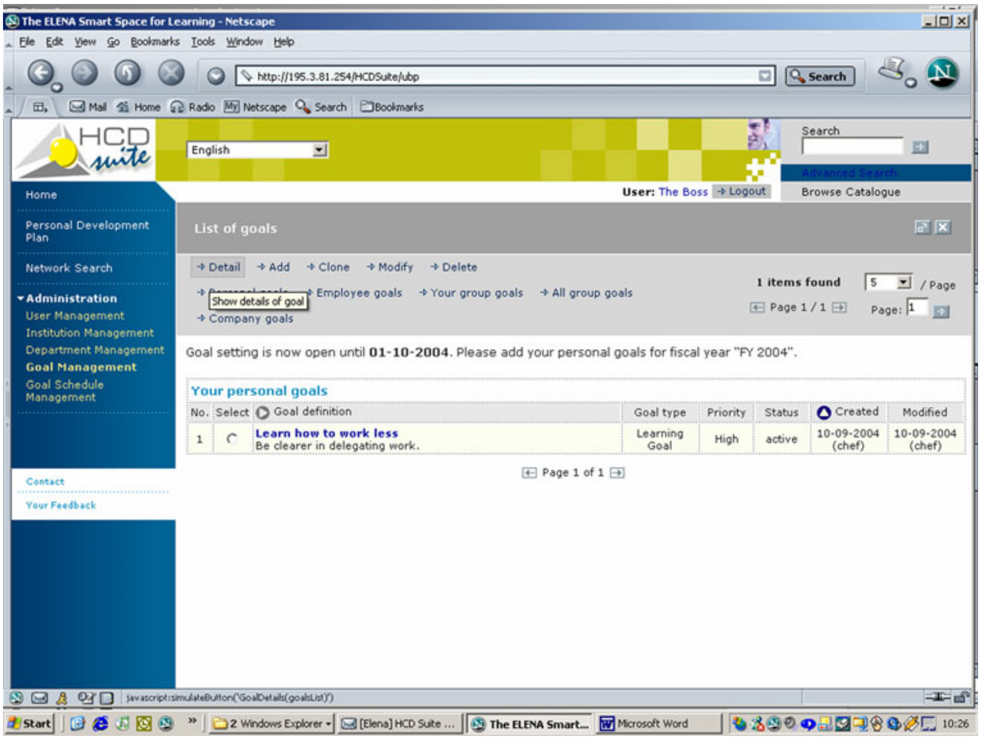
The effectiveness of techniques for requirements elicitation has been researched in a number of studies. Garmer et al. [20] conducted a case study of focus groups and compared the results with another study of focus groups and usability testing. Whereas the usability tests let users attend to the detailed requirements of the user interface, the focus groups enabled users to learn about contextual issues. Davis et al. [21] reported on a meta-analysis of 26 selected empirical studies using structured interviews, which appeared to be one of the most effective elicitation techniques. Methods such as card sorting or thinking aloud seemed to be less effective. They further reported that studies have not found the use of intermediate representations to have significant positive effects on elicitation. However, these findings were based on few empirical studies and in many cases have not been replicated.

Seyff et al. [22] investigated the effectiveness of scenarios on requirements discovery, comparing different forms of a scenario tool in scenario workshop walkthroughs, including the use of a scenario tool together with a first prototype of a learning system and creativity prompts. The results of this study showed that the session using a prototype produced more requirements than the non-prototype session. There is also further anecdotal evidence that prototypes are useful for gathering requirements and practitioners advice software companies to never visit a client without them [23].

Accepting the importance of the early discovery of the requirements, we sought to answer the following question:

> Is there a way to elicit the true requirements from future users of a system early in the software development cycle, or do the users need a high-fidelity prototype to be able to comment on missing, inadequate or incorrect features?

To further support the main question, additional questions were considered:



Goal setting is now open until **01-10-2004**. Please add your personal goals for fiscal year "FY 2004".

Your personal goals

No.	Select	Goal definition	Goal type	Priority	Status	Created	Modified
1	○	**Learn how to work less** Be clearer in delegating work.	Learning Goal	High	active	10-09-2004 (chef)	10-09-2004 (chef)

Fig. 1 A screenshot from a Hi-Fi prototype of the HCD-suite

1. Is a high-fidelity prototype necessary for users to convey missing, changed or excessive requirements of the system?
2. Is it possible to get as much feedback from participants who are not exposed to a conceptual model of the system through task practicing before an elicitation session?

2 Study design

This section describes the research study conducted to answer the questions put forward in the previous section. First, the domain of the study is explained. Next, the requirements elicitation technique, Scenario Acting, is described, and finally, the process and activities of the three groups that participated separately in the requirement elicitation process are outlined.

2.1 Learning with the human capital development suite

The study is about eliciting requirements for a system that will aid employees and managers in human capital development. This system, the HCD-suite, is innovative software that supports the training management life cycle, i.e. the steps that should be taken before, during and after a training measure. In addition, it distinguishes between the following four phases of the training management life cycle: (1) the learning goal definition and training needs analysis, (2) the selection and planning of training measures and the learning resource delivery, (3) coaching of the learning process and evaluation and (4) the transfer and outcome analysis. Typical tasks of this system would be, e.g., to register an employee's learning goals, find a course that meets these goals and an employee is interested in taking, register for a course and stay in touch with employee's managers and instructors of the course (Fig. 1).

The Learning Life Cycle, which is the basic concept of the HCD-suite design, is based on the work of Seeber [24]. The central design element of the HCD cycle is a workflow that engages potential learners, managers and human resources developers in a collaborative decision process to choose the right training measure. The goal-driven approach of the HCD-suite triggers the alignment of training selections with the explicit and implicit learning goals of individuals and organisations. Moreover, the HCD workflow triggers the evaluation of learning resources at the various stages of the HCD cycle, from the expectation analysis to the transfer analysis.

2.2 A requirements elicitation process

As Rudd et al. [25] pointed out, users may not be able to articulate their requirements, but they can talk about their goals and how they handle their tasks; from that, the system designer can find the solution that the user needs. From Svanæs and Seland [12] came the idea to have participants improvise scenarios and act them out. In the literature, the activity to engage users in creating scenarios and acting them out has been termed scenario building or role playing. In this paper, we have chosen to term it Scenario Acting to reflect the process described herein. The idea to conduct more than one Scenario Acting session came from Smith et al. [26], who asserted that users in a single session were more likely to focus on details rather than the big picture. Hence, in this study, a few days after a Scenario Acting session, a facilitator met individually with participants to inquire further about certain points of interest noted during the Scenario Acting session. This was a reflection exercise designed to get the most out of the acting sessions. Asking participants to log their activities was an attempt to get a better idea of what participants were thinking as they completed these tasks [27].

Participants in each group were divided into two teams consisting of two or three users. They were instructed to create a scenario related to several tasks and then act them out twice for the other team(s). During the first run-through of the Scenario Acting, the non-actors (audience) were encouraged to make notes of any questions or comments that they wanted to bring up during the second round of the performance. Then, the teams reversed roles; the audience turned to acting and the actors made notes. During the second round, the teams acted out their scenarios again. This time, participants were interrupted with questions and/or comments. Each Scenario Acting session took 90 min.

2.3 Experimental design: allocation of activities to groups

The research method applied was an experiment set in the field of a telecommunication company. Qualitative data were collected, and the researcher was assisted by two employees of the participating company, one from the Research Department and the other from Training Development. They served as liaisons between the researcher and the participants and probed the participants during the Scenario Acting sessions.

The study included fourteen participants, all employees of a telecommunication company, and their ages ranged from 24 to 60. Their job experience at the company varied, ranging from under 6 months to over 10 years. All the participants purported that their computer use at work was constant, and all but one claimed to use the Internet many times daily.

This study used three instruments: a high-fidelity, fully functional prototype; a low-fidelity paper prototype with an Internet browser for search tasks; and finally a written description of the system. The participants were divided into three groups (see Table 1): the Hi-Fi prototype group, the Lo-Fi prototype group and the control group. The first group, the Hi-Fi prototype group, was given access to a high-fidelity prototype of the HCD-suite system during a 2-week practice period. The participants were instructed to use this tool to enter their learning goals, book courses and then to have them accepted or rejected by their manager or supervisor. They were also asked to answer surveys offered by the system. None of the participants had used the Hi-Fi prototype prior to the experiment.

Table 1 Allocation of activities to groups

	The Hi-Fi prototype group	The Lo-Fi prototype group	The control group
Tasks introduced	×	×	×
Work on tasks, during practice time	×	×	
Use a working prototype	×		
Scenario Acting I	×	×	×
Scenario Acting II	×	×	
Reflection	×	×	

During a 2-week practice time, the employees in the Lo-Fi paper prototype group were instructed to write down their educational, personal and advancement goals using pencil and paper. The Lo-Fi prototype consists of paper forms, which included instructions for this. The forms supported the same tasks as the Hi-Fi group carried out, but the look of the interface was different. They then searched for courses to meet these goals and booked them with an Internet browser and by e-mail. (Note that the purpose of the Lo-Fi prototype is to serve as a low-level artefact to help participants engage in the tasks, but not to be a paper version of the future system's interface.) Finally, they had to keep track of the fulfilment of their goals and answer surveys. As we said above, both groups, Hi-Fi and Lo-Fi groups, had 2 weeks to work on these tasks, and they were asked to keep Activity Logs. The third group, the control group, was introduced to the features of the HCD-suite at the Scenario Acting session. This group did not carry out any of the tasks the other two groups did and thus did not receive any practice time. All participants were asked to comment on features that were either missing or irrelevant and were asked to suggest changes that could improve the system. Through this process, missing, irrelevant or changed requirements to the future system were gathered.

The study process, preparing participants for Scenario Acting, varied between the three groups. The Hi-Fi prototype and the Lo-Fi prototype groups participated in a preparatory meeting. During this meeting, each group was informed about the features of the system, using slides, and instructed how to complete the following tasks, which were in accordance with the process of the Learning Life Cycle:

1. Log onto the HCD-suite (only the Hi-Fi group)
2. Check the company's goals
3. Create personal goals
4. Search for courses
   a. that are available on the HCD-suite (only the Hi-Fi group) and
   b. on the Internet with a web browser
5. Sign up for courses and attach them to a goal
6. Have supervisor accept or reject a course
7. Participate in a course
8. Answer surveys on the course sent via e-mail

All participants, practicing tasks, were given forms to log their activities. For the Activity Logging, they were asked to approximate the time they took to complete the tasks, the type of activity performed, and any questions and/or insights they might have while completing that task. After 2 weeks, the first Scenario Acting session took place. Scenario Acting was introduced to participants through a demonstration of acting of a restaurant scenario unrelated to the human capital development application. Then, the facilitator asked the participants to act out scenarios involving the tasks listed above. The Scenario Acting session was followed by Reflection Meetings of 15–30 min with individual participants where the researcher asked participants to reflect on selected items from the Scenario Acting. A second Scenario Acting session took place 1 week after the first one.

The control group was treated differently. Participants of that group did not receive any practice time like the other two groups did, but had the features of the system explained to them. They received an overview of the features of the HCD-suite with slide presentations, and the following tasks were explained and discussed:

1. Analyse needs and define goals of learning
2. Plan and select training materials
3. Manage and evaluate learning
4. Analyse transfer and outcome of learning

This was followed by a Scenario Acting session, as with the two other groups.

2.4 Collection of the data

The qualitative data gathered from the participants were represented in three different ways: (1) Scenario Acting sessions, (2) Activity Logs and (3) Reflection Meetings. Activity Logs were diaries of activities kept by participants during the preparatory phase. The Scenario Acting sessions were videotaped, and written notes were made at the Reflection Meetings.

The statements made at the Scenario Acting sessions were transcribed and logged into a database along with statements collected from the Activity Logs and Reflection Meetings. The analyst (the first author of this paper) coded the statements in the database, with each code represented by one idea and identified by a sentence. Hence, one or more statement instances from the same group that had essentially the same meaning were coded with the same sentence. The coding for this study was done by creating first a coding scheme such that the first author coded parts of the data and discussed it with the second author. This was carried out iteratively. After the first coding scheme had been created, a second coding scheme was similarly created, the one which is reported in the paper. Moreover, an independent coder, not in the author group, has reviewed the coding, without seeing reasons for major changes.

3 Analysis and results

3.1 Frequency of requirements elicited

In analysing the results of this experiment, it was appropriate to note the total hours each group spent on the meetings and on the tasks during the preparatory period (Table 2). Because the participants did not always indicate the time spent on the tasks they logged, the overall time was underestimated.

The total number of statements per group for the groups Hi-Fi prototype, Lo-Fi prototype and control was 68, 74 and 33, respectively. The analyst uniquely filtered the statements into sentences within each group. Looking at the average number of statements per sentence, there was not much difference between the Hi-Fi prototype (1.77) and the Lo-Fi prototype groups (1.76), but the average was lower for the control group (1.22), which had only one Scenario Acting session.

Table 2 Time spent by participants in the three groups

	The Hi-Fi prototype group	The Lo-Fi prototype group	The control group
Number of participants	6	3	5
Time (hour:minutes)spent on tasks during practice period, according to a log	0 h:30 m	01 h:30 m	0 h:0 m
Total time spent on tasks and meetings	32 h	14 h:15 m	7 h:55 m
Average time spent per participant	5 h:20 m	4 h:45 m	1 h:35 m

When looking at the number of statements for the sentences of the three groups, the Lo-Fi prototype group had the most singleton sentences (i.e. one statement instance behind each sentence) or 26. Conversely, the control group had mostly singletons (23 out of 27). This was expected because the control group had only one Scenario Acting session. The Hi-Fi prototype group had the most sentences with multiple occurrences (i.e. fourteen twins, four triplets, one quadruplet and one quintuplet). However, an ANOVA showed that there was only a marginal statistically significant difference in frequencies between the groups, with $F(1,2) = 2.818$, $N = 107$ and $p = 0.06$. A further post hoc analysis with a Tukey's test revealed that there was a marginal difference in frequency between the Hi-Fi prototype and control groups ($p = 0.074$). The adjusted R^2 of 0.033 was very low, meaning that a group only contributed in a small way to the frequency of sentences.

3.2 New features, changes, observations and stated quality requirements

To determine whether different prototype fidelities and task practicing had an effect on the suggested requirements, the differences between the three groups with respect to the categories of new features, changes, observations or evaluations and stated quality requirements were analysed (Tables 3, 4). Whereas new features, changes and stated quality requirements can be translated into modified requirements to the system under development, observations are positive or negative statements about the system. We found that of the three groups, the Lo-Fi prototype group had the most suggestions for new features (51%) and the Hi-Fi prototype group had the highest number of changes (56%).

Table 3 Coding according to observations or suggested changes

Title of code	Explanation
New additional features	New feature is suggested
Major change	Suggestion implies major changes
Minor change	Suggestion implies minor changes
Observation	Observation of current work practice or feature of the system
Quality requirements stated	Stated requirements (e.g., regarding usability or performance)

Table 4 Changes, new features, observations and requirements

	The Hi-Fi prototype group	The Lo-Fi prototype group	The control group	Total
Changes	24% (9)	10% (4)	11% (3)	16
New additional features	24% (9)	48% (20)	37% (10)	39
Observations	53% (20)	36% (15)	52% (14)	49
Quality requirements	0% (0)	7% (3)	0% (0)	3
Total	100% (38)	100% (42)	100% (27)	107

An example of a new feature was (#174) 'a manager needs to advertise the courses which he wants to achieve', and an example of a change was (#55) 'important to translate into a native language'. In addition, the Hi-Fi prototype group made somewhat more observations (40% compared to 31 and 29% for the Lo-Fi prototype and the control groups, respectively).

The observations made by the participants were either negative or positive (e.g. negative: 'Can't book a course' or positive: 'Motivates me to think about why I want to take the course'). Examples of quality requirements were statements about usability and performance, such as 'The system should have low response time'. Interestingly, the Lo-Fi prototype group was the only group making quality requirements. Within each group, we saw that the majority of sentences within the Hi-Fi prototype and the control groups were observations (53 and 52% of the groups' sentences, respectively), but the Lo-Fi prototype group mostly suggested new features (48%). A chi-square analysis showed that the type of suggestions significantly depended on the type of group ($\chi^2 = 12.350$, $df = 6$, $p = 0.055$, $N = 107$).

We examined whether there was a large overlap of suggestions between the three groups (Fig. 2). The Hi-Fi prototype and the Lo-Fi prototype groups shared five suggestions for new additional features. There were three suggestions common to the Lo-Fi prototype and the control groups, but there were no suggestions on new additional features shared by the Hi-Fi prototype and the control groups.

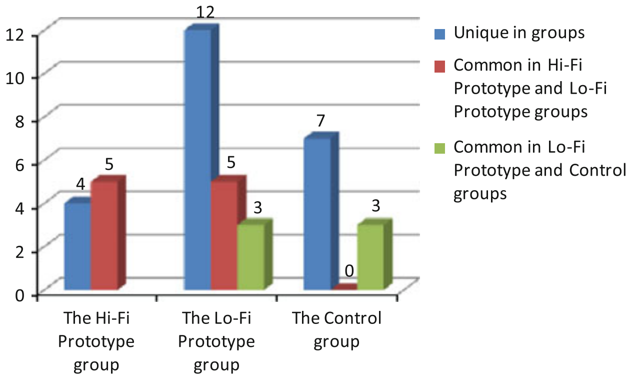


Fig. 2 Bar graph of new additional features of the three groups indicating both unique features and features suggested by more than one group

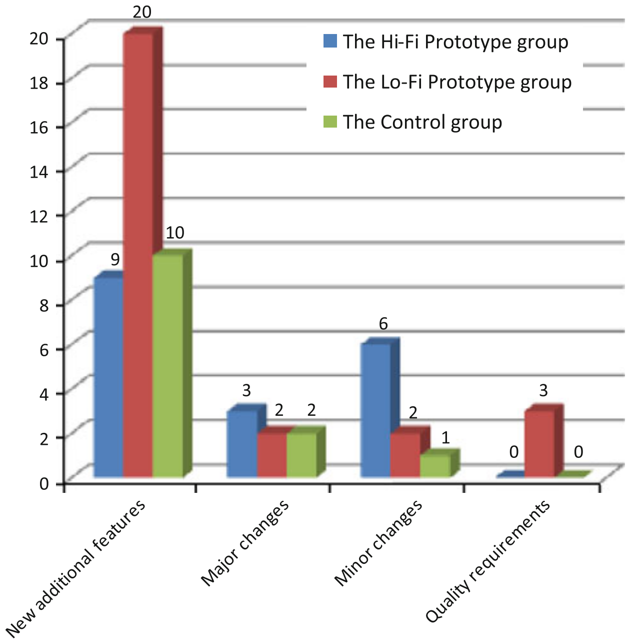


Fig. 3 Requirements—new additional features, changes and quality requirements

Finally, Fig. 2 shows that there was a greater number of unique new additional features suggested by the Lo-Fi prototype group (12) than by the control group (7).

Excluding observations, Fig. 3 shows the comparison of the three groups in terms of requirements made by the participants (i.e. the number of sentences coded as new additional features, major changes and minor changes and quality requirements). The Lo-Fi prototype group suggested more changes and additions than the two other groups, 24 compared to 18 and 13 by the Hi-Fi prototype and the control groups, respectively. When further distinguishing between minor and major changes, we saw that the difference between the groups lays in the number of minor changes, of which the Hi-Fi prototype group had the most, or 6.

It is notable how little difference there was between the two groups, Hi-Fi prototype and control, considering that participants of the Hi-Fi prototype group had two Scenario Acting sessions and 2 weeks to work on the tasks, whereas participants of the control group had only one Scenario Acting session and had the system described to them at the same session. A contributing factor was that the prototype limited the participants in the Hi-Fi prototype group because it hindered the participants' overview of the features. Apparently, neither practicing of tasks nor the second Scenario Acting session could compensate for the Hi-Fi prototype. Compared to the control group, more time was spent familiarising the Lo-Fi prototype group participants with the tasks of the system, and they spent time completing those tasks manually. In contrast, the participants of the control group did not have the advantage of having days to think about the tasks. Furthermore, the Lo-Fi prototype group participants had two Scenario Acting sessions and a Reflection Meeting in addition to their introduction meeting, while the control group participants had only one session combining the introduction with the Scenario Acting. It should also be noted that the Lo-Fi prototype group had fewer participants than the control group.

This analysis concluded that compared to neither having a high-fidelity prototype nor a practice period (the control group), a high-fidelity prototype with a practice period provided a higher number of minor changes and a higher number of observations. Comparing a low-fidelity prototype (i.e. paper and the Internet browser) to a high-fidelity prototype with a practicing period for both instruments, the low-fidelity prototype led to a larger number of suggested new features. Thus, to make the most of requirements gathering during the formative stage, a low-fidelity prototype with a practice period is desirable for generating both the most new features and the most new unique features.

3.3 Requirements related to the domain and system

Considering the suggested changes, additions and observations of the three groups, it was worthwhile to see whether there was any difference between the groups regarding the topics addressed by the participants. For example, it could be interesting to see whether one group discussed more domain-related topics, user interface-related topics, system-related topics or quality-related topics than another did. The sentences were labelled according to the codes shown in the first column of Table 5 with the qualitative analysis methods explained in Sect. 2.4. The table shows the number of sentences per code in each of the three groups. The final column of Table 5 shows a different broader category of the codes: *Practice of work, Ease of use and usefulness*, *Output and access* and *Learning process*. Table 6 presents an analysis of these categories across the three groups.

Table 6 shows that most of the sentences that were coded as usefulness or ease of use issues (19 occurrences, or 73% of all such sentences) came from the Hi-Fi prototype group. An example sentence coded as *New-ease-of-use* was 'configure a page to include what I use most' or 'a user can configure the page as he likes best' (#40). In comparison, there were four sentences in this category in the Lo-Fi prototype group and only three in the control group. The three sentences from the control group categorised in *Usefulness* and *Usefulness+* codes were participants' observations of seminars that they had participated in or heard of. These were not system related (e.g. 'employees need more encouragement to book a seminar'). Moreover, participants from the Hi-Fi prototype group had

Table 5 Coding relating to features—cells with more than four sentences are in bold

Code name	Description of code	Number of sentences per group			Category
		Hi-Fi group	Lo-Fi group	Control	
Current practice	How things are practiced now	3	**6**	3	Practice of work
Ease of use −	Negative comments on ease of use	**5**			Ease of use and usefulness
Ease of use +	Positive comments on ease of use	1			Ease of use and usefulness
Feature-access	Already a feature of the system relating to access	1	3	1	Output and access
Feature-evaluation	Already a feature of the system relating to evaluation of learning		2	3	Learning process
Feature-goal setting	Already a feature of the system relating to goal setting of learning	3	1	2	Learning process
Feature-request	Already a feature of the system relating to requests learning	1	2		Learning process
Feature-search	Already a feature of the system relating to searching for courses	1	2	2	Learning process
Future practice	How work should be practiced in the future independent of the system		1	2	Practice of work
New-access	New feature relating to access		**4**		Output and access
New-ease of use	New feature relating to ease of use	**4**	**4**		Ease of use and usefulness
New-evaluation	New feature relating to evaluation	1	2	**5**	Learning process
New-future practice	New feature relating to future practice	2	**6**	**4**	Practice of work
New-goal setting	New feature relating to goal setting	**5**	**5**	1	Learning process
New-output	New feature relating to output		3		Output and access
New-search	New feature relating to search	2	1	1	Learning process
New-usefulness	New feature relating to usefulness	2			Ease of use and usefulness
Usefulness	Comment on usefulness	1		2	Ease of use and usefulness
Usefulness −	Negative comment on usefulness	**4**			Ease of use and usefulness
Usefulness +	Positive comment on usefulness	2		1	Ease of use and usefulness

the prototype to deal with and they, understandably, were focused on the details of the usability and ease of use of the system. Thus, we concluded that the prototype tended to draw the participants' attention away from missing or inadequate requirements because they focused on dealing with the system itself.

Looking only at those topics that were related to the activities of the learning process, such as goal setting, searching, requesting a course and evaluation, there was not much difference in numbers between the groups (13, 15 and 14 occurrences for the Hi-Fi prototype, Lo-Fi prototype and control groups, respectively). However, proportionally within the group, the control group commented most on *Learning processes*. An example of a sentence in *New-goal setting* was 'throw out the goals when they have been achieved 100%' (#89). In a category related to the learning processes, the category of *Practice of work*, the Hi-Fi prototype group had three sentences, the Lo-Fi prototype group had seven sentences, and the control group had five sentences. A typical example of a sentence in this

Table 6 System-oriented versus domain-oriented sentences

	The Hi-Fi prototype group	The Lo-Fi prototype group	The control group	Total
Practice of work	13% (5)	31% (13)	33% (9)	25% (27)
Ease of use and usefulness	50% (19)	10% (4)	11% (3)	24% (26)
Output and access	3% (1)	24% (10)	4% (1)	11% (12)
Learning process activities: goal, search, request, evaluation	34% (13)	36% (15)	52% (14)	40% (42)
Total in groups	100% (38)	100% (42)	100% (27)	100% (107)

category was ‘The system needs to manage job descriptions’. Finally, when considering topics related to output and access, e.g. *feature-access*, *new-access* and *new-output*, which we classified as system-related topics, the Hi-Fi prototype and the control groups stated one sentence each and the Lo-Fi prototype group stated ten. An example of such a sentence was ‘Need to be able to print information, a list of goals, courses, etc’.

As mentioned previously, the sentences were coded into the categories described in the last column of Table 5, i.e. *Practice of work*, *Learning process*, *Ease of use and usefulness* and *Output and access*. We classified the two former categories, *Practice of work* and *Learning process*, as related to the domain, while the two latter categories, *Ease of use and usefulness* and *Output and access*, were related to the system. An overview of the aforementioned analysis is provided in Table 6. A chi-square analysis revealed that the topic of the sentence depended strongly on the group ($\chi^2 = 30.745$, $df = 6$, $N = 107$ and $p = 0.000$).

This analysis concluded that the Hi-Fi prototype group is mostly focused on ease of use and usefulness and outnumbers by far the other two groups. As expected, a high-fidelity prototype is needed to motivate discussion on ease of use. The Lo-Fi prototype group more or less evenly divided its comments across *Practice of work*, *Learning process activities* and *Output and access*. Of the three groups, this group had far the most comments in that last category. This might have been because they used a web browser during the practice time. The control group focused most suggestions on *Learning process activities*, followed by *Practice of work*. A conclusion can be that the less system-related artefacts a group has, the more focus the discussion is on domain-oriented topics.

3.4 Domain- and system-related requirements and type of statements

In this section, we studied the relationship between the topic code introduced in the previous section and the new features, changes and observations discussed in Sect. 3.2. The goal was to see if the topics determined whether sentences were suggestions of changes, additional features, observations or quality requirements. We examined the relationship between the four topics (*Practice of work*, *Ease of use and usefulness*, *Output and Access* and *Learning process*) and the three groups of Hi-Fi prototype, Lo-Fi prototype and control within each code type of changes, new additional features, observations and quality requirements. The results of a chi-square analysis showed that there was not a statistically significant dependency between topics and groups within the suggested changes ($p = 0.113$), but there was clearly a significant dependency between topics and groups within new additional features ($\chi^2 = 17.728$, $df = 6$, $N = 39$, $p < 0.05$) and within observations ($\chi^2 = 15.221$, $df = 6$, $N = 49$ and $p < 0.05$).

The Hi-Fi prototype group made most of its observations (see Fig. 4) on the topic of *Ease of use and usefulness* (11 out of 20). Of the three groups, the Lo-Fi prototype group was the frontrunner in observing *Practice of work,* and proportionally, the control group observed *Learning process activities* the most.

Taking only sentences that suggested new features (see Fig. 5), the sentences of the Hi-Fi prototype group were divided among *Learning process activities* and *Ease of use and usefulness.* What is interesting is that the Hi-Fi group can suggest new additional features for *Learning process activities*. The Lo-Fi prototype and the control groups discussed new features of *Practice of work*, and

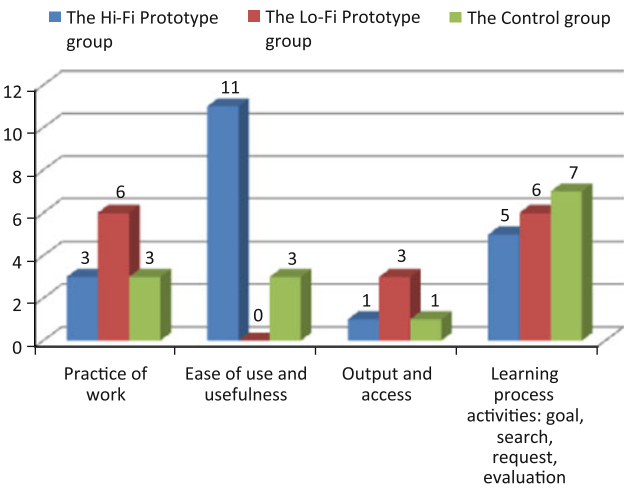


Fig. 4 Observations—topics by group

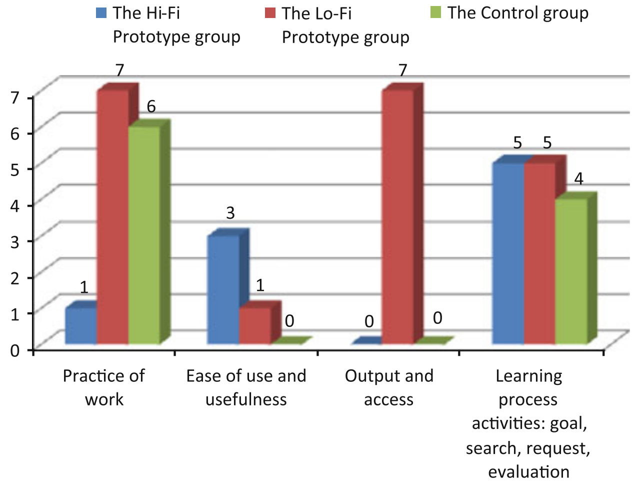


Fig. 5 New additional features—topics by group

additionally, the Lo-Fi prototype group was the only group that suggested new features for the topic of *Output and access*.

3.5 Repetitions between Scenario Acting sessions

To determine whether participants behaved differently in the two sessions, I and II, we further analysed the data from the Hi-Fi prototype and Lo-Fi prototype groups. Table 7 shows the distribution of the unfiltered statements among Scenario Acting sessions in the two groups. In the first Scenario Acting session, and the only one for the control group, the participants voiced 33 sentences. The Hi-Fi prototype group had slightly more sentences in the first scenario session (54%) than in the latter. Conversely, the Lo-Fi prototype group had considerably more sentences in the second scenario session (68%).

Because we were only concerned with groups that had two sessions, we eliminated the control group from the following analysis. The goal was to see whether it was worthwhile to hold the second session (i.e. how many additional sentences were expressed in the second session and whether there was a difference between the groups or a difference in topics between sessions). Table 8 shows that the Lo-Fi prototype group was slower in getting started and definitely needed the second session, with only 34% of the sentences expressed during the first session. Conversely, it seemed that the Hi-Fi prototype might have helped participants of that group express requirements immediately in the first session.

Table 7 The groups' unfiltered statements in sessions I and II

	The Hi-Fi prototype group	The Lo-Fi prototype group	The control group
Session I	54% (37)	32% (24)	100% (33)
Session II	46% (31)	68% (50)	0
Total	100% (68)	100% (74)	100% (33)

Table 8 Groups by session—filtered sentences

First expressed in/group	The Hi-Fi prototype group	The Lo-Fi prototype group
Session I	66% (25)	26% (11)
Session II	34% (13)	74% (31)
Total	100% (38)	100% (42)

Table 9 Topics discussed in sessions I and II

	Session	The Hi-Fi prototype group	The Lo-Fi prototype group	Total
Practice of work	I	3	5	8
	II	2	8	10
	Total	5	13	18
Ease of use and usefulness	I	11	0	11
	II	8	4	12
	Total	19	4	23
Output and access	I	1	3	4
	II	0	7	7
	Total	1	10	11
Learning processes	I	10	3	13
	II	3	12	15
	Total	13	15	28

To determine whether session I was sufficient, we analysed the topics of the sentences discussed in session II that had not been discussed in session I (Table 9). An analysis of the topics discussed in the session for each group showed a statistically significant dependency for the *Learning process* sentences between groups and sessions, with the Lo-Fi prototype group expressing more sentences on *Learning process* in the latter session (twelve in the Lo-Fi prototype group vs. three in the Hi-Fi prototype). However, the Hi-Fi prototype group expressed more sentences in the former session (ten in the Hi-Fi prototype group vs. three in the Lo-Fi prototype group). A chi-square analysis revealed a significant difference in discussing the *Learning process* in sessions between the groups ($\chi^2 = 9.073$, $df = 1$, $N = 28$ and $p = 0.003$).

3.6 Summary of results

The frequency of the sentences depended marginally significantly on the groups. The control group, in which only one session was conducted, had the lowest frequency.

The type of suggestions (new features, changes or observations) depended significantly on the groups, with the Lo-Fi prototype group suggesting the most new features and the Hi-Fi prototype group making the most observations. The Lo-Fi prototype group suggested the most unique features.

The topic of the sentences was significantly dependent on the groups. The Hi-Fi prototype group discussed *Ease of use and usefulness* the most, while the Lo-Fi prototype group discussed *Output and access* and *Practice of work* the most. The control group focused most of its suggestions on *Learning process activities* followed by *Practice of work*.

A further breakdown of the topics by the categories of new features and observations revealed that the control group discussed *Practice of work* and *Learning processes* when suggesting new features. The Lo-Fi prototype group followed a similar pattern in addition to commenting on *Output and access.* Finally, observations made by the Hi-Fi prototype group were on *Ease of use and usefulness*.

Analysing the sessions (I or II) in which the Hi-Fi prototype and Lo-Fi prototype groups voiced their sentences, the Lo-Fi prototype group was the slow starter, with more sentences at the later session. In contrast, the Hi-Fi prototype group had more sentences in the first session. Finally, the Lo-Fi prototype group discussed the *Learning process* more in the second session than in the first.

4 Discussion

The purpose of this experiment was to find out whether there was a way to elicit requirements from future users of a system early in the software development cycle or whether they needed to have a high-fidelity prototype. Could users who only had the features of the system described to them produce as many new or changed requirements as users who were able to work on a prototype to familiarise themselves with the system? We can conclude that the users who had the system described to them without the aid of the high-fidelity prototype were able to produce a greater number of new or changed requirements than users who worked with it. In this respect, working with the prototype was more of a hindrance for the users. The users with the Hi-Fi prototype got so involved with the details of the system that they got distracted from contemplating new or changed features. Their comments were focused on issues such as ease of use and usefulness more than those of the other groups.

We expected that the prototype would force the participants of the Hi-Fi prototype group to focus on working with the system, which would cause them to be less focused on what could be changed or added. The data supported this theory, at least partially. The collected data showed that participants of the Lo-Fi prototype group produced more of the new additional features, and the participants of the Hi-Fi prototype group produced more minor changes than participants in the other groups did. This supported the expectation that participants using the Hi-Fi prototype tended to be less focused on additional or different features for the system. We also expected the control group to produce fewer overall changes, which they did, but we expected the difference to be greater.

We expected that there would be a greater difference between the groups in terms of how much they discussed the *Learning process*, but the results showed that overall the control group had slightly more than expected sentences in this category and consistently for the individual codes of changes, new additional features and observations. The difference lays more in the *Output and access* topic, which the Lo-Fi prototype group discussed more often, and in the topic of *Practice of work*, for which the Lo-Fi prototype and the control groups had more suggestions of new features. The Hi-Fi prototype group was most concerned about *Ease of use and usefulness.* Broadly speaking, we concluded that those groups having some type of prototype were more focused on system-related issues, such as output and access and ease of use and usefulness than those not having a tool.

Regarding the question of whether it is necessary to create a working prototype before involving the users in the requirements elicitation process, it seems that the users can contribute to the requirements without the aid of the prototype. In fact, the high-fidelity prototype was found to be a poorly suited tool for obtaining requirements from users. This study also found that a paper prototype of a system and search tool, such as the tool used by the participants of the Lo-Fi prototype group, had great value in early requirement gathering.

The analysis of repeated sentences between sessions revealed that there was definitely a benefit in holding a second session. The Lo-Fi prototype group was a slow starter and needed the second session to discuss the majority of the requirements. For the *Learning processes* in particular, the Hi-Fi prototype group was quicker in getting to this subject in the first session.

Another question we pondered regarded the time spent with the users who did not work with the high-fidelity prototype (i.e. Lo-Fi prototype and control groups). The users who were given time to work on the tasks of the system for 2 weeks and also had two Scenario Acting sessions and a Reflection Meeting (the Lo-Fi prototype group) produced considerably more new and changed requirements than the group that only had one meeting (the control group). Therefore, we concluded that spending time with the users and giving them time to familiarise themselves with the system produced better results. This was in agreement with work published previously by Garmer et al. [20]. To see further whether this was due to the practicing of the tasks or the second Scenario session, we looked at the overall sentences by sessions. The data indicated that practicing the tasks was actually a hindrance for the Lo-Fi prototype group as they produced only 11 sentences during the first session compared to 27 of the control group. It would be worthwhile to repeat this study in the future by planning two scenario sessions for each group Lo-Fi (with task practicing) and control (without task practicing). As we see from this study, one of the difficulties in doing studies of task practicing is that the time participants report on the practicing of tasks may be underestimated.

5 Validity and reliability of the results

5.1 Internal validity

When we assessed the validity of the research design, we determined that two items could be a threat to the internal validity of the results. First, there was not an equal number of participants in each group. Having fewer participants in one group could either have the effect of having fewer sentences, if we assumed that sentences are voiced per person, or it could have the effect of more sentences because of the social effect between the participants. However, we divided each group into teams and each team performed in turn, which could be seen as mitigating this risk. A second threat to validity was that for the control group, two conditions were changed (i.e. the participants could not practice, and they only had one session to discuss the requirements of a future system). Thus, it might be difficult to deduce whether their behaviour, if different from the other groups, was due to the lack of practicing tasks or to the single session.

Another possible threat to validity may be the small number of participants and their selection. The participants of this study were recruited from a group of participants undergoing continuing education in a company. They were randomly selected to the three groups. The main danger is that the number of sentences produced by the participants was too small and somehow biased. A statistical power analysis [28] showed that the sample allowed us to detect differences if they are moderate, as the effect size for number of sentences, $N = 107$ was $w = 0.36$ with alpha $= 0.05$ and beta equal to 0.20. (A small effect size is $w = 0.1$, moderate is $w = 0.3$, and large effect is $w = 0.5$.) [29, 30]. Thus, in cases where no difference between groups was detected, we cannot exclude that if these differences were small, our sample was not able to detect them. This is especially true for Sect. 3.4 when we compare the topics of the different groups within the type of suggestions where the number of sentences in each category is much lower. The low frequency of the sentences in Sect. 3.1, i.e. number of statements behind each sentence, can indicate that the study may have benefitted from having more teams for each group. Although not an uncommon frequency pattern in studies comparable to this one [31], having more teams could result in better convergence of sentences. Also, since the work is performed in groups, team members may influence each other, especially with dominant members, and there can be variations between groups. These problems are to some extent also found in focus groups and have been address in the literature [32]. However, the method Scenario Acting secures that all participants actively participate, since the facilitator ensures that everyone takes turn in acting out the scenarios and commenting as part of the audience to the acting group. Furthermore, the disciplined method of coding the communication into sentences and statements ensures a methodological way of analysing the data.

5.2 External validity

The answer to the question of whether we can generalise the findings of the experiment is the same in this study as with any contextual experiment: There are always environmental factors that must be considered. In particular, two factors should be mentioned that could threaten external validity, the domain of the study and the expertise of the participants. Although the study has been applied in the particular domain of learning, the results can probably be extended to other domains, which apply similar processes of goal setting, search, selection and evaluation or other processes in the corporate context. Concerning the second factor, then the participants were typical corporate users in the domain, neither experts in requirements elicitation nor learning. Neither of these factors are major threats but are raised here to show possible limitation to those wanting to apply the methods in other domains that have very different characteristics from the one used in this study.

5.3 Reliability of coding

Coding of statements is subjective to the researcher analysing the data. The coding in this study was carried out according to grounded theory, i.e. analyst reads the data and identifies codes that will help him or her understand it and associate meaning to it. Hence, the coding scheme is not prepared beforehand but emerges from the data. This is a subjective process and never a mechanical one. It is meant to increase the analyst's understanding of the data, help him or her derive knowledge from it, but the resulting coding scheme will depend partly on his or her objectives of the study and on own knowledge of the subject. A number of factors influence its reliability, such as the analyst's experience of the domain and his or her fluency with qualitative analysis methods. A few techniques can be used to decrease the unreliability of the coding. One is to have two or more researchers develop the code, thus doing a pilot study with parts of the data, reaching an agreement and iterating this process until all the data have been coded. It should be noted that this will always be a collaborative effort and only done independently by the coders for parts of the data. A second method is a member check, i.e. by asking the interviewee about the coding and receiving a confirmation from them [33]. Third is to ask the coder to recode the initial data to examine the intra-coder agreement. This study has used the first method above.

6 Conclusion

At the onset of this study, we asked whether users would need a high-fidelity prototype to discover requirements and whether they would benefit from preparing for a Scenario Acting session by practicing the tasks to be realised by the system. From this study, we conclude that users who have the system described to them with the aid of a low-fidelity prototype are able to come up with a greater number of new or changed requirements than users who work with a high-fidelity prototype. Working with the Hi-Fi prototype appeared to be a hindrance for users, as they become so involved with the details of working the system that they became distracted from contemplating new or changed features that it may have been appropriate for them to suggest.

This study also tried to determine the value of having users familiarise themselves with the tasks of the system in question. For this purpose, the Lo-Fi prototype and control groups were compared. The users who were given time to work on the tasks of the system for 2 weeks and had two Scenario Acting sessions and a Reflection Meeting (the Lo-Fi prototype group) produced considerably more new and changed requirements than the group who did not practice the tasks and only had one meeting (the control group). Further work could include research into whether this was due to practicing of tasks or the second Scenario Acting session.

In addition to the findings on the usefulness of different prototype fidelities and the efficiency of task practicing, the method of user involvement applied by the Lo-Fi prototype group demonstrated our contribution to devising a viable method of involving users in establishing requirements. When implementing this experiment, we devised a method to involve potential users in establishing the requirements for a system early in the design cycle. As with all methods, their application will spur ideas for improvement as future work. At the Scenario Acting sessions, we emphasised that the participants were not to take themselves too seriously during these sessions but that they should try to have fun and play to encourage creativity. For example, Maiden et al. [34] used balloons at their workshops to provoke play and interaction. Producing colourful, fun objects, seemingly unrelated to the topic of the system, to include in the scenarios could help the participants generate ideas. Another valuable addition to the method would be to ask the participants to rank the sentences that were produced. For example, they might be asked which sentences were highly valued or less valued. The technique of Action Research [35, 36] may be suitable for improving the method devised in this paper. When applying Action Research, an experiment with an intervention is conducted, and then the process is evaluated and improved, if necessary, before an improved experiment is conducted. Thus, a plan, action, observe, reflect cycle is conducted, and the researcher is a part of the development team. System developers may have a different view of the usefulness of Scenario Acting and low-fidelity prototypes, as has been found in [13] where developers explained that role play and low-fidelity prototypes could not be used alone. Thus, action research is likely to reveal the strengths and the weaknesses of the method from the system developers' perspective, which will complement this study.

References

1. Charette RN (2005) Why software fails [software failure]. IEEE Spectr 42(9):42–49
2. El Emam K, Koru AG (2008) A replicated survey of IT software project failures. IEEE Softw 25(5):84–90
3. Lamsweerde A (2000) Requirements engineering in the year 00: a research perspective. In: Ghezzi C, Jazayeri M, Wolf AL (eds) Proceedings of the 22nd international conference on software engineering. ACM, Limerick, Ireland, pp 5–19
4. Kamata MI, Tamai T (2007) How does requirements quality relate to project success or failure? In: Sutcliffe A, Jalote P (eds) Requirements engineering conference, 2007. RE'07. 15th IEEE international, Delhi, pp 69–78
5. Boehm B, Papaccio P (1988) Understanding and controlling software costs. IEEE Trans Softw Eng 14(10):1462–1477
6. Preece J, Rogers Y, Sharp H (2002) Interaction design—beyond human–computer interaction. Wiley, New York
7. Cheng BHC, Atlee JM (2007) Research directions in requirements engineering. In: Briand LC, Wolf AL (eds) Future of software engineering, 2007. FOSE'07. IEEE Computer Society Washington, Washington, DC, pp 285–303
8. Olsson E, Johansson N, Gulliksen J, Sandblad B (2005) A participatory process supporting design of future work, vol 2005-018. Uppsala University, Uppsala
9. Kujala S (2003) User involvement: a review of the benefits and challenges. Behav Inf Technol 22(1):1–16

10. Ehn P (1988) Work-oriented design of computer artifacts. Arbetslivscentrum Stockholm, Sweden
11. Iacucci G, Iacucci C, Kuutti K (2002) Imagining and experiencing in design, the role of performances. In: Proceedings of the second Nordic conference on human–computer interaction. ACM, Aarhus, Denmark, pp 167–176
12. Svanæs D, Seland G (2004) Putting the users center stage: role playing and low-Fi prototyping enable end users to design mobile systems. In: Dykstra-Erickson E, Tscheligi M (eds) Proceedings of CHI. ACM, Vienna, Austria, pp 479–486
13. Seland G (2006) System designer assessments of role play as a design method: a qualitative study. In: Morch A, Morgan K, Bratteteig T, Ghosh G, Svanæs D (eds) Proceedings of the 4th Nordic conference on human–computer interaction: changing roles. ACM, Oslo, Norway, pp 222–231
14. Suri JF, Marsh M (2000) Scenario building as an ergonomics method in consumer product design. Appl Ergon 31(2):151–157
15. Lim Y-K, Pangam A, Periyasami S, Aneja S (2006) Comparative analysis of high- and low-fidelity prototypes for more valid usability evaluations of mobile devices. In: Morch A, Morgan K, Bratteteig T, Ghosh G, Svanæs D (eds) Proceedings of the 4th Nordic conference on human–computer interaction: changing roles. ACM, Oslo, Norway, pp 291–300
16. Hall R (2001) Prototyping for usability of new technology. Int J Hum Comput Stud 55(4):485–501
17. Luqi VB, Guan Z, Berzins V, Zhang L, Floodeen D, Coskun V, Puett J, Brown M (2004) Requirements-document-based prototyping of CARA software. Int J Softw Tools Technol Transf (STTT) 5(4):370–390
18. Alavi M (1984) An assessment of the prototyping approach to information systems development. Commun ACM 27(6): 556–563
19. Tudhope D, Beynon-Davies P, Mackay H (2000) Prototyping praxis: constructing computer systems and building belief. Hum Comput Interact 15(4):353–383
20. Garmer K, Ylvén J, Karlsson M (2004) User participation in requirements elicitation comparing focus group interviews and usability requirements for medical equipment: case study. Int J Ind Ergon 33:85–98
21. Davis A, Dieste O, Hickey A, Juristo N, Moreno AM (2006) Effectiveness of requirements elicitation techniques: empirical results derived from a systematic review. In: Glinz M, Lutz R (eds) Requirements engineering, 14th IEEE international conference. IEEE, Minneapolis/St. Paul, MN, pp 179–188
22. Seyff N, Maiden N, Karlsen K, Lockerbie J, Grünbacher P, Graf F, Ncube C (2009) Exploring how to use scenarios to discover requirements. Requir Eng 14(2):91–111
23. Schrage M (2004) Never go to a client meeting without a prototype. IEEE Softw 21(2):42–45
24. Seeber S (2000) Stand und Perspektiven von Bildungscontrolling. In: Seeber S, Krekel EM, van Buer J (eds) Bildungscontrolling. Ansätze und kritische Diskussionen zu Effizienz-Steigerung von Bildungsarbeit, Frankfurt am Main, pp 19–49
25. Rudd J, Stern K, Isensee S (1996) Low vs. high-fidelity prototyping debate. Interactions 3(1):76–85
26. Smith H, Fitzpatrick G, Rogers Y (2004) Eliciting reactive and reflective feedback for a social communication tool: a multisession approach. In: Designing interactive systems. ACM, Cambridge, MA, USA, pp 39–48
27. Hornbæk K, Frøkjær E (2004) Two psychology-based usability inspection techniques studied in a diary experiment. In: Proceedings of the third Nordic conference on Human–Computer Interaction (NordiCHI'04). ACM, Tampere, Finland, pp 3–12
28. Faul F, Erdfelder E, Lang A-G, Buchner A (2007) G*Power 3: a flexible statistical power analysis program for the social, behavioral, and biomedical sciences. Behav Res Methods 39(2): 175–191
29. Kampenes VB, Dybå T, Hannay JE, Sjøberg DIK (2009) A systematic review of quasi-experiments in software engineering. Inf Softw Technol 51(1):71–82
30. Cohen J (1988) Statistical power analysis for the behavioral sciences. Laurence Erlbaum, Hillsdale
31. Hvannberg ET, Law EL-C, Lárusdóttir MK (2007) Heuristic evaluation: comparing ways of finding and reporting usability problems. Interact Comput 19(2):225–240
32. Klein EE, Tellefsen T, Herskovitz PJ (2007) The use of group support systems in focus groups: information technology meets qualitative research. Comput Hum Behav 23(5):2113–2132
33. Taylor SJ, Bogdan R (1998) Introduction to qualitative research methods. Wiley, New York
34. Maiden N, Gizikis A (2004) Provoking creativity: imagine what your requirements could be like. IEEE Softw 21(5):68–75
35. Baskerville R, Pries-Heje J (1999) Grounded action research: a method for understanding IT in practice. Account Manage Inf Technol 9(1):1–23
36. Avison DE, Lau F, Myers MD, Nielsen PA (1999) Action research. Commun ACM 42(1):94–97