The Role of Requirements Engineering Practices in Agile Development: An Empirical Study

Xinyu Wang[1], Liping Zhao[2], Ye Wang[3], and Jie Sun[4]

[1] College of Computer Science, Zhejiang University, China
wangxinyu@zju.edu.cn
[2] School of Computer Science, The University of Manchester, Manchester, U.K.
liping.zhao@manchester.ac.uk
[3] School of Computer Science and Information Engineering, Zhejiang Gongshang University, China
yewang@zjgsu.edu.cn
[4] Ningbo Institute of Technology, Zhejiang University, China
jiangbin@zju.edu.cn

Abstract. Requirements Engineering (RE) plays a fundamental role in all sorts of software development processes. Recently, agile software development has been growing in popularity. However, in contrast to the extensive research of RE in traditional software development, the role of RE in agile development has not yet been studied in depth. In this paper, we present a survey with three research questions to explore the treatment of RE in the practical agile development by investigating eight agile groups from four software development organizations. To answer the three research questions, we targeted at 108 participants with rich agile experiences and designed a questionnaire to collect their answers. Our survey shows that agile RE practices play a crucial role in agile development and they are an important prerequisite for projects' success though many agile methods advocate coding without waiting for formal requirements and design specifications.

Keywords: requirements engineering, agile software development, scrum, survey, requirements analysis.

1 Introduction

In recent years, agile software development grows in popularity as it tackles a lot of software development problems in dynamic contexts, such as Scrum [1], XP [2] and so on. The RE process runs through the whole agile development process [3]. Many RE practices have been proposed particularly for agile development [4][5]; however, little literature studies the role of agile RE practices and the attention paid to agile RE in practice from the empirical perspective.

In order to move a step towards understanding the role of RE practices in agile development, we conducted an empirical study to explore and explicate the RE process during a project in the context of agile software development. To ensure rigor, we designed this study by strictly following the survey process proposed in [6]. We have

D. Zowghi and Z. Jin (Eds.): APRES 2014, CCIS 432, pp. 195–209, 2014.

set out to answer the following research questions (RQs) through the survey in eight agile groups.

RQ1: Is agile RE performed as important as what agile practitioners thought?

RQ2: Which type of requirements (i.e. FRs and NFRs) are agile practitioners more concerned with?

RQ3: At each RE stage, what methods and tools do agile practitioners often use?

For RQ1, we attempt to investigate whether agile RE is performed different from what agile practitioners expected and what consequences it brings. RQ2 investigates if non-functional requirements (NFRs) are treated as important as functional requirements (FRs) in agile development. Finally, RQ3 aims to investigate what methods and tools agile practitioners usually use for requirements elicitation, requirements representation and documentation, and requirement management. In a word, RQ1 studies the overall role of agile RE played in agile development. RQ2 studies agile RE from the problem aspect whereas RQ3 studies agile RE practices from the solution aspect.

In the following sections, we will present the whole process of the survey and the conclusion we have obtained.

2 Research Method

2.1 Survey Design

Sampling. Before conducting a survey, we need to choose a relevant survey population. According to [7], “a prerequisite to sample selection is to define the target population as narrowly as possible”. In this survey, therefore, the target population should be those who have participated or are participating in agile projects. In order to make the survey results as precise as possible, we selected agile practitioners from four different kinds of software development organizations which have established collaborations with the authors.

The four organizations that we chose have their own typical characteristics. The first company (C1) is the worldwide leader in networking that transforms how people connect, communicate and collaborate. It provides various kinds of products and services such as borderless networks, data center and virtualization, VOIP phones and gate-way systems, video conferencing and so on. The second company (C2) is a software development firm that specializes in information processing and management. It provides services to financial and health organizations worldwide. The third company (C3) is a multinational corporation that designs, develops and manufactures flash memory storage solutions and software. The fourth company (C4) is a software outsourcing company in China. It develops a large amount of services systems and provides different kinds of services for its customers around the world. We selected the survey sample from the software developers who either have been involved or are involved in agile projects.

According to the ability of the interviewer to gain access to the study subjects, we chose four agile groups (G1, G2, G3, G4) from C1, one group (G5) from C2, two groups (G6&G7) from C3 and one group (G8) from C4. The total number of participants from these groups is 108. Table 1 presents the detailed information of these eight groups is listed in whereas table 2 explains the primary role that the survey participants had in the studied projects.

Table 1. Participant and project details. Project duration is in week

Group ID	Group Members	Agile Method	Project Description	Domain	Project Duration	No. of Iteration
G1	Master x 1 BA x 1 Dev x 2 QA x 1	Scrum	Web Conference System Development	Network	40	20
G2	Master x 1 BA x 1 Dev x 3 QA x 2	Scrum	Online Training System Development	Network	28	14
G3	PO x 1 Designer x 1 Dev x 3 QA x 2	Scrum	Survey Report System development	Network	36	12
G4	PO x 1 Dev x 3 QA x 2	Scrum	Integration of Client Support Systems	Network	26	13
G5	Master x 1 BA x 2 Designer x 1 Dev x 6 QA x 4	Scrum	Fund management system reengineering	Finance	96	24
G6	Coach x 1 Designer x 1 Dev x 8 QA x 5	XP	SQL Database Development	Database	24	12
G7	Coach x 1 BA x 1 Designer x 2 Dev x 14 QA x 9	XP	Flash Optimization	Memory Storage	24	8
G8	Master x 1 BA x 3 Designer x 1 Dev x 13 QA x 9	Scrum	Barreled Water Ordering Management System Development	E-commerce	15	5

Survey Types Selection. There are many types of surveys, such as written survey, face-to-face survey, phone survey and mixed mode survey. Our survey is a mixed mode survey, which consists of two parts, including both face-to-face survey and written survey.

First, we had a face-to-face interview with agile coaches (Scrum Masters or XP Coaches), POs and BAs of each agile project on the development process of each project, specifically on the RE process. Then we sent our electronic questionnaire to all participants for their responses. The reason why we chose these two survey modes is that it would take a lot of time to face-to-face interview and impossible to let all group members involve in the interview. Another reason is that not all group members know well about the RE process. Therefore, it is not necessary to let all of them involve in the first-part interview.

Table 2. Participants summary in the survey

Participants' primary role	No. of participants
Agile Coach	6
Designer	6
BA	8
PO	2
Developer	52
QA	34
Total Number of Participants	108

2.2 Survey Instrument Development

At a fundamental level, one challenge in developing survey instrument is to design a coherent set of questions. As said by Fowler [8], "a good question is one that produces answers that are reliable and valid measures of something we want to describe".

In order to design an applicable questionnaire, we followed a set of three steps as proposed in [9]:

1) Compose a preliminary questionnaire which is relevant to the aforementioned three RQs,
2) Do a pilot interview to agile coaches, POs and BAs of each group and ask them to validate the questionnaire,
3) Implement changes in the questionnaire based on their feedback.

The questionnaire consists of both open-ended questions and close-ended questions. Open-ended questions allows the participants to freely present what they think about the RE practices in agile development, whereas close-ended questions require the respondent to choose from among a given set of responses, such as those provided by Likert scales. Compared to open-ended questions, close-ended questions are easiest for participants to answer and for researchers to analyze the data. Therefore, in order to save participants' time, close-ended questions account for more than 50% in our questionnaire.

2.3 Survey Execution

Face-to-Face Interview. We conducted open interviews with agile coaches, POs and BAs from the eight groups.

First, we asked agile coaches to give us a general introduction to their development process and then directly asked them what kind of problems they have encountered in their projects. Each interview session was between 1 hour and 2 hours. This interview provided an overview for us to understand how each project executed and what kind of role agile RE played at each stage.

Second, we asked BAs to describe agile RE process in their projects and what kind of problems they have encountered during agile RE. As some POs were also responsible for requirements analysis, we also interview POs in this round interview. We use an umbrella name "Requirements Analyst (RA)" for both BAs and POs. The interview questions focused on the participants' experiences of working with agile RE practices and in particular around their roles on agile projects. For example, we asked about the challenges RAs faced in projects and the strategies they used to overcome them. The answers varied with the individual participants. This process usually took 30 to 60 minutes.

Throughout each interview session, the interviewer took handwritten notes. These notes helped us to discover questions from what interviewees stated.

Finally, the interviewer and the interviewee walked through the questionnaire which served to guide the second-part survey.

Written Survey. The questionnaire was composed of four parts, each of which corresponds to a research question except the first part. The first part aims to investigate the background of the participant including his/her role in the agile project, project size, agile method and so on. Answers in this part have been summarized in Table 1 and Table 2.

The second part, corresponding to RQ1, consists of four questions: the first question discussed the importance of agile RE that each participant considered whereas the second question discussed the actual effort paid by the participants with respect to the RE practices of agile projects. The third and fourth question investigated the problems or difficulties that different groups have encountered. The third part corresponds to RQ2 and discusses the degree of participants' concern on different types of requirements in different agile development phases. The last part corresponds to RQ3 and includes questions related to the agile RE methods and tools. Questions in the last part can be categorized into four classes: requirements elicitation, requirements representation and documentation, requirements analysis and requirements management. The final questionnaire is shown in Table 3. Some questions have multiple options.

3 Results

Table 3. Questions in the survey. Questions specific for RAs are marked as *

Questions
RQ1: Is agile RE performed as important as what agile practitioners thought?
Q1: How important do you think agile RE is?
Q2: How many RAs are there in your project?
Q3: What's the percentage of delayed iterations in your project?
Q4: Which development activity(s) usually causes difficulties in your project?
RQ2: Which type of requirements (i.e. FRs and NFRs) are agile practitioners more concerned with?
Q5: Which type of requirement costs more effort in your project?
Q6: How much attention is paid to FRs in the following development activities – project planning, requirements analysis, design, coding, and testing? Please sort the five activities from the most to the least.
Q7: How much attention is paid to NFRs (e.g. performance, security, reliability, etc) in the following development activities – project planning, requirements analysis, design, coding, and testing? Please sort the five activities from the most to the least.
Q8: Are customers satisfied with the quality of the system your group developed?
RQ3: At each RE stage, what methods and tools do agile practitioners often use?
Q9: How often do you communicate with customers to discuss requirements?*
Q10: What type of method do you use to elicit requirements? *
Q11: Why do you choose the above method(s) to elicit requirements?*
Q12: Do you use any tools to elicit requirements? What are they?*
Q13: What type of methods do you use to describe requirements? *
Q14: Do you use any tools to document requirements? What are they? *
Q15: What method do you use to find any inconsistency, incorrectness or incompleteness in the requirements you captured from the customer? *
Q16: Do you pay a lot of attention to the dependency among requirements, such as the dependency between two FRs or between NFRs? If yes, how do you maintain these dependencies? *
Q17: What method do you use to manage requirements changes? *
Q18: Why do you choose the above method to manage requirements changes? *
Q19: Do you use any tools to manage requirements? What are they? *

3.1 RQ1: Is Agile RE Performed as Important as What Agile Practitioners Thought?

According to the 108 responses, we obtained the answer for Q1 as shown in Figure 1. More than 90% participants considered RE important, amongst which 85.19% considered it very important. None of the participants denied its importance. Therefore in most agile practitioners' minds, agile RE is undoubtedly crucial for the success of their projects despite their role.

To investigate how many efforts have been devoted in agile RE practices, we designed Q2 and Figure 2 shows the result of this question in groups. All of G1, G2, G3, G4, G6 and G7 had one RA, whereas G5 assigned two RAs and G8 assigned three. Noted that for G3, G4, G6 and G7, they didn't particularly assign one person to do the

RA job; instead, their coach plays the role of RA. The percentage of RAs in total number was from 3.7% to 20%. Most groups severely cut the resources for agile RE for the purpose of cost saving. In the face-to-face interview, we learned that all agile coaches preferred to assign more resources to coding and testing though they have encountered many problems during requirements analysis. We can see that most agile RE does not get due attention as it was thought before.

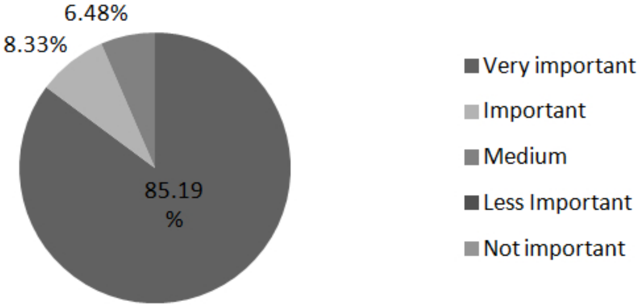

Fig. 1. The importance of RE in agile practitioners' minds

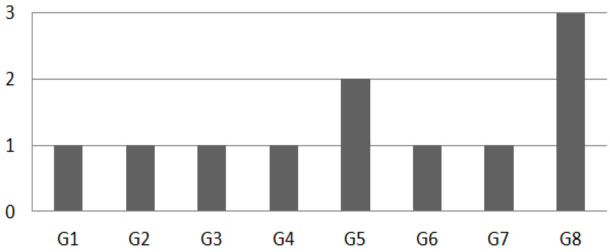

Fig. 2. The number of people assigned to RE in agile projects

In order to obtain the delay rate of each group, we designed Q3 and the results are shown as Fig. 3. Of all the eight groups, G6, G7 and G8 have much higher delay rates which respectively are 50%, 41.67% and 40%, follow by G3, G4 and G5 whose delay rates respectively are 19.44%, 21.43% and 25%. G1 and G2 have the lowest delay rates which respectively are 5% and 7.69%.

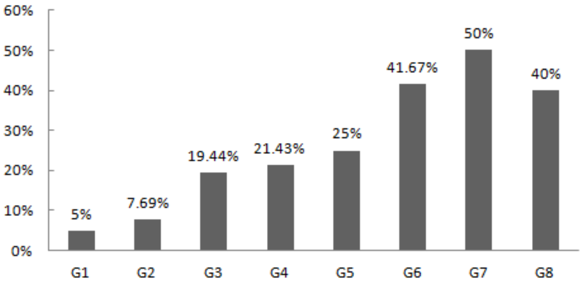

Fig. 3. The delay rate of each group

Figure 4 shows the reasons that cause delays in the previous question. In all the five development activities, requirements analysis is considered by 78 participants as the most crucial factor that causes delays, followed by the factor of project planning as preferred by 42 participates. The other three development activities only receive small amounts of blame: 16 participants ascribe delays to the testing stags; 13 participants ascribe delays to design; 7 participants ascribe delays to coding.

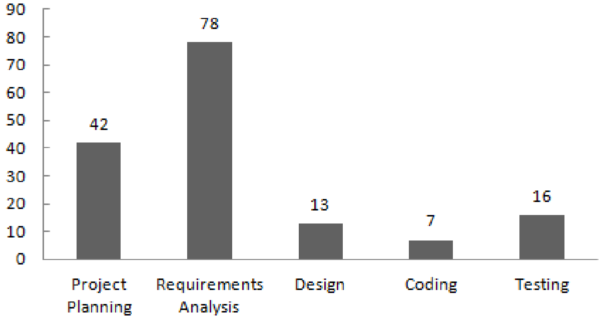

Fig. 4. The delay rate of each group

3.2 RQ2: What Type of Requirements Are Agile Practitioners More Concerned With?

We categorized requirements into two major categories: FRs and NFRs. We found that the majority of groups (75%) treated FRs more important than NFRs in the context of agile development, whereas only 25% of groups treated them as equally important, as shown in Figure 5.

We observed that the attention paid to FRs varies in both development activities (i.e. project planning, requirements analysis, design, coding and testing) and groups. The participants listed five activities in an order of most to least attention that have been paid to FRs, which is shown in Table 4. Most groups assessed efforts in terms of function points (FPs) of system requirements; as a result, they paid most attention to FRs at the beginning of the project, i.e. at the stage of project planning. There is no doubt that almost all groups put "Requirements Analysis" in the first or second place as the aim of this activity is to analyze FRs. Table 4 tells us that every group paid less attention to FRs when coding and testing.

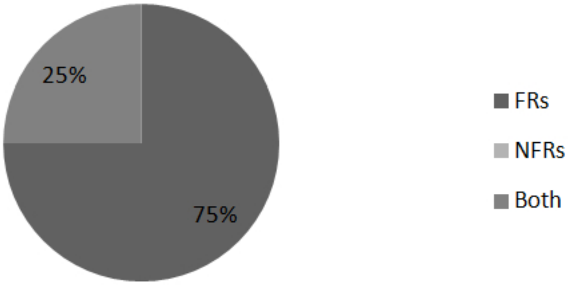

Fig. 5. NFRs compared to FRs

Table 4. The degree of attention paid to FRs in terms of development phases and groups

Group ID	Attention paid to FRs in the order of development phases
G1	Project Planning, Requirements Analysis, Testing, Design, Coding
G2	Requirements analysis, Design, Project Planning, Coding & Testing
G3	Requirements Analysis, Testing, Design, Project Planning, Coding
G4	Project Planning, Requirements Analysis, Testing, Design, Coding
G5	Requirements analysis, Project Planning, Design, Testing, Coding
G6	Project Planning, Requirements Analysis, Design, Testing, Coding
G7	Project Planning, Requirements Analysis, Design, Coding, Testing
G8	Project Planning, Requirements Analysis, Design, Coding, Testing

For NFRs, the degree of attention is significantly different from what was obtained for FRs. Table 5 shows the result. For example, they paid more attention to NFRs when designing and testing than that they paid to FRs. Most groups didn't consider NFRs at the beginning of their projects as their customers only concerned FRs; instead, these groups started to deal with NFRs from the stage of design. Testing is the only way to evaluate whether the system meets NFRs, therefore, it is also necessary to pay a lot of attention to NFRs during testing.

Table 5. The degree of attention paid to NFRs in terms of development phases and groups

Group ID	Attention paid to NFRs in the order of development phases
G1	Requirements Analysis, Testing, Design, Coding, Project Planning
G2	Project Planning, Design, Testing, Requirements analysis, Coding
G3	Requirements Analysis, Testing, Design, Coding, Project Planning
G4	Project Planning, Requirements Analysis, Design, Coding, Testing
G5	Design, Requirements analysis, Project Planning, Coding, Testing
G6	Design, Requirements Analysis, Testing, Coding, Project Planning
G7	Testing, Requirements Analysis, Project Planning, Design, Coding
G8	Design, Coding, Testing, Project Planning, Requirements Analysis

To obtain the customer satisfaction of system quality, we designed Q8. The results are as follows: Customers of G1, G3 and G4 are highly satisfied with their system quality; Customers of G5, G6 and G7 are satisfied with their system quality; finally customers of G2 and G8 are dissatisfied with their system quality.

3.3 RQ3: What Methods and Tools Do Agile Practitioners Often Use?

Questions for RQ3 are only left particularly for RAs to answer, so as to conduct an in-depth investigation on RE practices including both methods and tools.

Requirements Elicitation. In agile groups, 45.45% of RAs discussed requirements with customers once a week, 20.37% twice a week, 16.67% communicated with customers once they needed (more frequent than once a week) and only 9.09% communicated with their customers every day (see Figure 6).

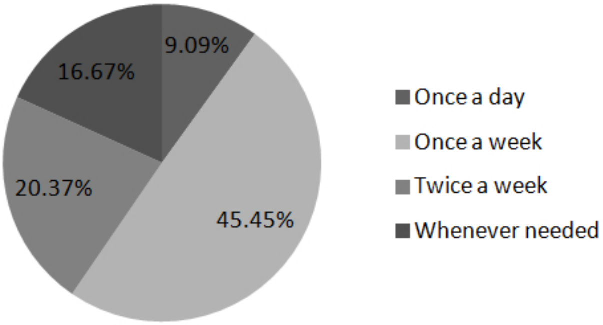

Fig. 6. The frequency of requirements elicitation from customers

We also investigated the common methods that were used by RAs to elicit requirements and the reason why RAs preferred those methods. From Figure 7, interview and user story were the most frequently used methods (More than 90% RAs used them). As interview is as means of face-to-face communication with customers, RAs found it "*quite simple and efficient to capture requirements from customers*". Compared to other methods, interview is more efficient and direct; moreover, it is easy to use for most RAs. Since most customers cannot define their requirements correctly, RAs adopted user story to "*refine vague requirements into more precise and detailed ones*".

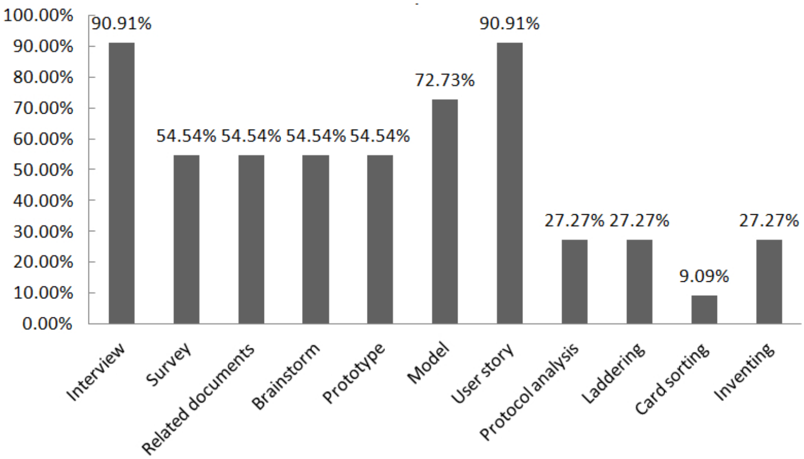

Fig. 7. The requirements elicitation methods used by RAs

In many RE surveys, modeling was usually mentioned by RE practitioners to elicit requirements, such as business process modeling [10], goal modeling [11]. In our survey, modeling was the second frequently used method for requirements elicitation. 72.73% of RAs preferred to use models as "*models can show their ideas easily and quickly*".

Face-to-face survey [12] and prototyping [13] were also preferred by a lot of RAs due to the ability that they can "*enable in-depth understanding between stakeholders*". Brainstorming was popular in agile groups as it can "*involve as many stakeholders as possible to participate in the requirements discussion and clarification*". As G5 aimed to reengineer a legacy system, analyzing existing documents was the most important way to retrieve requirements.

Compared to the aforementioned approaches, knowledge acquisition methods [14] such as protocol analysis, laddering, card sorting and inventing requirements were less often used by RAs. The reason is that only a few RAs are familiar with these methods.

Based on the above methods, three types of tools were often used by RAs: 1) Microsoft Word and Excel were used for interviewing, user story authoring and survey; 2) Visual [15], Enterprise Architect [16] and other UML modeling tools were used when RAs wanted to build models; 3) Balsamiq Mockup [17] and Photoshop [18] were used for prototyping.

Requirements Representation and Documentation. According to our interview to the RAs, we summarized six frequently used methods for representing and documenting requirements. The usage percentage of each method is shown in Figure 8.

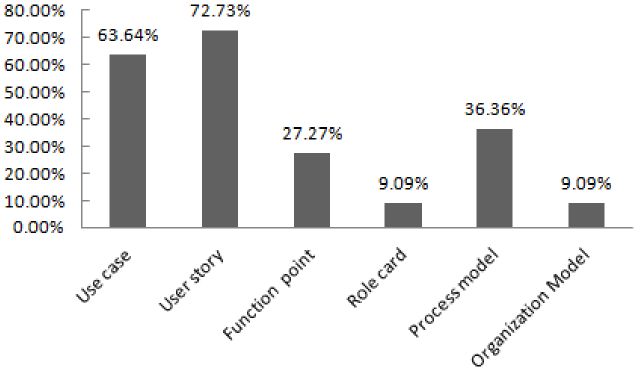

Fig. 8. The requirements representation methods used by RAs

This figure reveals that both use case and user story are the most preferred methods. This is not surprising because these two methods are very well known due to their prominent comprehensibility, decomposability and interactivity. Additionally, function point and process model are also frequently used for specific intensions. For example, function point is capable of expressing system internal functions; process model is a necessity for describing the requirements of process-intensive systems. Sometimes, role card and organization model are also used as the peripheral support for requirements representation and documentation.

To facilitate the representation and documentation of requirements, tools are commonly leveraged. The most popular ones are Microsoft Word and ScrumWorks [19]. Others such as Wikipage [20], Testlink [21], Microsoft Excel and Microsoft Visio are also utilized by some QAs as auxiliary tools.

Requirements Analysis. Another responsibility of RAs is to discover and handle inconsistent, incorrect and incomplete requirements (i.e. requirements defects) captured from customers. According to the responses from RAs, there are several ways helping them to discover requirements defects.

For example, the common method they used is comparison. They designed a set of scenarios and then compared the process flows of these scenarios to check the inconsistency and incompleteness between requirements. Another way is to confirm the existing requirements with customers and experts. Generally, customers are more sensitive to requirements incorrectness than BAs as they are more familiar with the business. However, some requirements defects cannot be discovered at the stage of requirements analysis. In some cases, RAs asked developers and QAs for their helps to find out these defects.

Requirements Management. In agile software development, the changes of requirements and the dependencies between requirements are so frequent that they need to be maintained and managed.

Requirement changes are usually managed by several ways in agile groups. For example, BAs from G5 managed changes through highlighting them in the requirement documentations and also maintained documentation versions through SVN [22]. In doing so, they make requirements easy to trace and the whole process more time-saving. The BA from G8 managed changes through building a change table, which consisted of details and effects of requirement changes, project risks, solution plans. All of these items would be confirmed by relevant stakeholders. Other project groups employed some requirements management tools to assist to manage changes, such as JIRA [23], Rally [24] and ScrumWorks [19].

In addition to requirement changes, it is crucial to identify and manage the dependencies between requirements, as the dependencies between requirements determine the order to implement different functionalities. According to the questionnaire, most groups take actions to handle requirement dependencies except G2. Both G1 and G8 mentioned that they tried to decrease the dependencies between requirements through coupling business-relevant requirements together. G3, G4 and G5 marked the dependencies through requirements prioritization. G6 and G7 managed the dependencies through building a two dimensional matrix, whose column and row were both requirements.

4 Discussion and Threats

4.1 Discussion of Results

Little literature studies agile RE practices. Our study looked at agile RE practitioners from three different aspects and revealed a few findings as below:

1. With respect to the overall project management aspect, we observed that the importance of agile RE practices deviated from what agile practitioners thought in their minds. Although the majority of agile groups acknowledged the importance of RE in agile development, very few resources were allocated to capture and analyze customers' requirements, which results in constant requirement changes from

customers. Correlating to the interview with agile coaches from each group, we found that project delays usually occurred in G6, G7 and G8 and their agile coaches admitted that the vague, incomplete and ambiguous requirements were the major reason for the project delays, whereas in other groups, project delays were less common. It is interesting to find that the proportion of the number of RAs in these groups was between 14% and 20% while the proportion in G6, G7 and G8 was under 10%. This study suggests that agile practitioners should carefully leverage the time cost and labor cost of RE in their projects.

2. With respect to the problem aspect, the study found that most agile practitioners were more concerned with FRs than NFRs. Only a few part of them treated FRs and NFRs equally. The major reason was that customers usually cannot give an accurate definition of NFRs. They didn't know how they expected the system to be performed. They cannot tell agile developers where they did not satisfied until they started to use the system. As a result, agile practitioners didn't treat NFRs seriously in the first couples of iterations. It is interesting to notice that FRs got more attention at the stage of project planning and requirements analysis whereas NFRs were just the opposite, i.e. at the stage of design and testing. However, the quality of systems developed by those groups that didn't pay much attention to NFRs at the stage of requirements analysis was not satisfying. The quality issues were hurriedly fixed in the last several iterations according to the feedback from customers. This finding suggests that agile practitioners should not ignore NFRs analysis at the beginning of their projects.
3. With respect to the solution aspect, we observed that in agile practices, most RAs discussed requirements with customers very often, about 80% of RAs got in touch with their customers at least once a week, 100% at least twice a week. Two reasons led to this situation: 1) In each iteration, new requirements need to be implemented and it is necessary to confirm new requirements with customers before implementation; 2) RAs wanted to capture requirement changes as early as possible in order to reduce the total cost brought by project changes. Interview, user story, modeling and prototyping are the most widely used requirements elicitation methods, whereas user story and use cases are the most widely used requirements representation methods. We also observed that both customers and RAs preferred to use natural-language-based methods to describe requirements due to the fact that such methods are simple and intuitive. Yet, these methods will generate a large number of documents which make them difficult to manage. This finding suggests that agile RE practices should promote requirements management tools in order to manage a large set of requirements, prioritizations and the dependencies between requirements.

4.2 Threats to Validity

We evaluated the possible threats to validity of the results we have obtained from the empirical study.

First, the level of agile practitioners' experience is one possible threat which may affect the results. We believe this threat to validity is small as we selected experienced agile practitioners from different agile groups. All these agile practitioners have participated in at least one agile project. Second, we acknowledge the possible risks that answering all questions may be time-consuming for the participants. In order to save their time, we substituted a lot of close-ended questions for open-ended questions. However, we admit that there is another threat as we were noticed that close-ended questions may restrict the scope of answers to those questions. Besides, the list of questions we present is not comprehensive, which needs improvement in the future.

5 Conclusions

Our study reveals that RE practices play a crucial role in agile development. Although many agile methods advocate coding without waiting for formal requirements and design specifications, RE is still an important prerequisite for the success of projects. The value of the work presented in this paper is the identification of a set of following findings about RE practices for agile practitioners.

Lack of concern on RE in practice. Many project delays were caused by insufficient communication with customers or shortage of RE resources. The role of agile RE practices has been acknowledged by most agile practitioners, nevertheless, they didn't make an even resource allocation due to the project cost. Therefore, it is recommended that agile groups should carefully leverage the resources assigned to RE and the costs.

Lack of concern on NFRs. Agile practitioners should pay more attention to NFRs at the beginning of the project, rather than leave them to design or coding. We suggest them to adopt some methods such as NFR framework for NFRs analysis.

Preference for agile RE practices. In agile RE practices, the participants identified that interviewing and user story are the most important requirements elicitation practices whereas user story and use case are the most widely used requirements representation practices. In order to rapidly capture requirement changes from customers, it is important to intensively communicate with customers. Although agile RE differs from traditional RE in that it takes an iterative discovery approach, many traditional RE practices are still applicable in agile development such as use case, modeling and prototyping and so on.

Acknowledgements. This work has been supported by the National Natural Science Foundation of China under Grant No. 61103032. We also would like to thank those who participated in our study and gave us a lot of valuable feedbacks.

References

1. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall PTR (2001)
2. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley (1999)

3. Paetsch, F., Eberlein, A., Maurer, F.: Requirements Engineering and Agile Software Development. In: 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 308–313. IEEE Press, New York (2003)
4. Cao, L., Ramesh, B.: Agile Requirements Engineering Practices: An Empirical Study. IEEE Software, 60–67 (January/February 2008)
5. Ramesh, B., Cao, L.: Agile Requirements Engineering Practices and Challenges: An Empirical Study. Information Systems Journal 20, 449–480 (2010)
6. Glasow, P.: Fundamentals of Survey Research Methodology (2005), `http://www33.homepage.villanova.edu/edward.fierros/pdf/Glasow.pdf`
7. Salant, P., Dillman, D.A.: How to Conduct Your Own Survey. John Wiley and Sons (1994)
8. Fowler, J., Floyd, J.: Improving Survey Questions: Design and evaluation. Sage Publications (1995)
9. Racheva, Z., Daneva, M., Sikkel, K., Wieringa, R., Herrmann, A.: Do we Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study. In: 18th International Requirements Engineering Conferences, pp. 147–156. ACM Press, New York (2010)
10. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 30–49. Springer, Heidelberg (2000)
11. Rolland, C., Souveyet, C., Achour, C.B.: Guiding Goal Modeling Using Scenarios. IEEE Transactions on Software Engineering 24, 1055–1071 (1998)
12. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap. In: 4th International Conferences on Software Engineering, pp. 35–46. ACM Press, New York (2000)
13. Davis, A.: Operational Prototyping: A New Development Approach. IEEE Software 9, 70–78 (1992)
14. Shaw, M., Gaines, B.: Requirements Acquisition. Software Engineering Journal 11, 149–165 (1996)
15. Visual Paradigm: Visual Paradigm for UML (2013), `http://www.visual-paradigm.com/`
16. Sparx Systems: Enterprise Architects (2013), `http://www.sparxsystems.cn/products/ea/`
17. Balsamiq. Balsamiq Mockups (2013), `http://balsamiq.com/products/mockups/`
18. Adobe Systems: Adobe Photoshop (2013), `http://www.adobe.com/cn/products/photoshop.html`
19. CollabNet: ScrumWorks Pro (2013), `http://www.collab.net/downloads/scrumworks`
20. Wiki Books. Wiki Page (2013), `http://en.wikibooks.org/wiki/Wikipage`
21. Syncro Soft.: TestLink (2013), `http://testlink.org/`
22. Apache.: Subversion (2013), `http://subversion.apache.org/`
23. Atlassian.: Jira (2013), `https://www.atlassian.com/zh/software/jira`
24. Rally Software: Rally (2013), `http://www.rallydev.com/`