Innovations Syst Softw Eng (2014) 10:235–241
DOI 10.1007/s11334-014-0232-4

ORIGINAL PAPER

State of practice in requirements engineering: contemporary data

Mohamad Kassab · Colin Neill · Phillip Laplante

Received: 16 February 2014 / Accepted: 4 April 2014 / Published online: 16 April 2014

Abstract Little contemporary data exists that documents software requirements elicitation, requirements specification, document development, and specification validation practices. An exploratory survey of more than 3,000 software professionals was conducted and nearly 250 responses were obtained. Survey data obtained includes characteristics of projects, practices, organizations, and practitioners related to requirements engineering. Selected results are presented along with interpretations of this data.

Keywords Requirements engineering · Common practices · Requirements specification · Software development industry · Software professionals · Requirements elicitation

1 Introduction

In 2003 the authors [1] presented results of a comprehensive survey showing requirements engineering practices across a broad range of industries and projects types. The results were surprising in that they indicated, among other things, that the Waterfall model was still widely used and that various techniques associated with Agile development were not widely employed. Results from a similar survey in 2008 [2] indicated that the findings from 2003 had remained largely unchanged. The 2003 survey results were highly cited (181 times via Google Scholar), and seemed to provide a template for more focused requirements surveys. For example Khurum et al. [3] conducted a brief survey to uncover challenges in organizations to effective requirements engineering, Chernak [4] surveyed a small group of companies to determine the prevalence of requirements reuse and Verner et al. [5] sought to uncover specific issues with respect to requirements management. But since 2003 and again in 2008 the results of no other comprehensive surveys of requirements engineering practices were published.

To remedy this deficiency, and provide useful data to other researchers, we updated and reprised the 2003 and 2008 surveys. The present survey includes responses from a broader geographic base including international participants. Selected data from this most recent survey is exhibited herein along with some interpretations of the meaning of this data.

2 Survey design and conduct

We created a Web-based survey using the Web-based QuestionPro survey tool (http://www.QuestionPro.com). The survey consisted of 32 questions (summarized in Table 1). Participants were drawn from multiple sources:

1. A database of former students in the Masters of Software Engineering degree program at the Penn State Great Valley School of Graduate Professional Studies (PSGV). PSGV caters primarily to working professionals. An email invitation (and subsequent reminder) was sent to these individuals.
2. Subscribers to the IEEE Reliability Society newsletter, which has a circulation of >3,000 professional members. An email invitation and reminder was sent to the associated listserver.

M. Kassab (✉) · C. Neill · P. Laplante
Engineering Division, The Pennsylvania State University,
30 E Swedesford Rd, Malvern, PA 19355, USA
e-mail: muk36@psu.edu

C. Neill
e-mail: cjn6@psu.edu

P. Laplante
e-mail: plaplante@psu.edu

Table 1 Summary of survey questions

No.	Question	Type
1	Which of the following categories best describes the types of software developed for this project?	MCN
2	Which of the following application domains does/did this project apply to?	MCN
3	What is/was the duration of the project (from inception to delivery)?	MC1
4	If you are following/followed an iterative development for this project, how many iterations (sprints) is/was the project spread through?	MC1
5	On Average, what is/was the duration (in business days) for one iteration (sprint) in this project?	D
6	On Average, how much technical debts are/were carried from one iteration (Sprint) to the next?	MC1
7	How many full time staff (IT) are/were involved in the project altogether?	D
8	How would you estimate the size of the project in terms of lines of code?	MC1
9	Which of the following development lifecycles best describes the one you are using/did use in the project?	MC1
10	Which of the following Integrated Application Lifecycle Management (ALM) tools was/is used for this project?	MC1
11	Within the lifecycle, do/did you do any prototyping?	MC1
12	If your answer is Yes, how do/did you prototype?	MCN
13	What techniques do/did you use for requirements elicitation?	MCN
14	Which of the following approaches are you using/did use in analysis and modeling the software requirements?	MCN
15	In what sort of notation is/was the requirements specification expressed?	MC1
16	Do/Did you or your team perform requirements inspection for this project?	MC1
17	If your answer is Yes, which technique do/did you use?	MCN
18	Do/did you or your team perform estimation for the size of requirements or the effort of building them?	MC1
19	If your answer is Yes, which technique do/did you use?	MCN
20	Were Non-Functional Requirements considered during the size/effort estimation?	MC1
21	Which of the following Software Quality Management approaches best describes the one you are using/did use in the project?	MC1
22	The following statements are indicators for Software Quality and Software Productivity (Please rate these statements by clicking one box with the following scales)	LS
23	Please answer the following questions with regard to "this project":	SDS
24	Please answer the following questions with regard to "your company":	SDS
25	In your opinion, does your company...	SDS
26	Which of the following describes your position while engaged in this project?	MC1
27	Over the last 5 years, how many software projects have you worked on?	D
28	What type of business or organization are/were you employed by during this project?	MC1
29	Approximately how many software professionals are/were employed by your organization?	MC1
30	In what country is/was your organization located?	DDM
31	In what US region is/was your organization located?	MC1
32	What is/was the approximate size of your organization's annual budget?	MC1

D dichotomous, *LS* Likert scale, *MC1* multiple choice (single selection), *MCN* multiple choice (multiple selection), *SDS* semantic differential scale, *DDM* drop-down menu

3. Members of the following Linked-In professional groups, to which one or more of the authors belonged: Requirements Engineering Specialist Group (RESG), Computing Reviews Reviewers, CSIAC Software Intensive Systems Engineering (formerly DACS), DAMA (data managers association)/Philadelphia region, DAMA (data managers association)/International, IEEE Computer Society members, Software Engineering, IEEE—USA. An invitation to participate was posted to these groups.

Survey data was collected from April 2013 through July 2013. The survey drew 247 participants from 23 countries. Of these survey takers; 119 completed the survey. The completion rate was 48 % and the average time taken to complete the survey was 15 min. We also included the results of the partially completed responses. When respondents aborted the survey, they tended to do so on or near question 23, we speculate from "survey fatigue".

3 Participant and project characteristics

The survey drew responses from professionals across a wide range of industries. Responses from the education industry were 13.5 % of the total volume followed by responses from the banking and finance industry with 13 and 11 % from the IT industry.

The survey participants also reflected a diverse range of positions describing themselves as programmers/developers, software/system engineers, or testers 61 % of the time. Architects, project/product managers, analysts, and consultants comprised the remaining 39 % of respondents; positions typically involved in the higher-level aspects of computerized system's technical design. Given this population, responses to the survey are more likely to reflect the opinions and biases of any given project's development team rather than those of other groups represented in a software development effort.

As might be expected from such a diverse population, respondents also reported a wide range of project experience. When asked how many projects they had worked on in the last five years answers ranged from two to more than 50. The mean experience level in this survey in terms of number of projects in the last 5 years is 22.57; and the median experience level is 17.5. The standard deviation of 16.25 is indicative the broad spectrum of experience levels reported. Nevertheless, respondents were asked to base all their project responses on one project only that they were either currently involved with or had taken part in during the past 5 years.

Several questions asked respondents to classify and categorize the relevant project. In general, projects were distributed across a broad range of application domains with a mild bias towards applications in the banking/financial sector (22 % reported this domain). The projects were also distributed across different categories with a bias towards Web-based projects (50 % reported this type).

With respect to project schedules, the majority of the projects (59 %) were a year or less in duration. For those projects that followed an Agile development method (e.g. Scrum, Extreme Programming, Feature Driven Development etc.), 43 % took <10 iterations to complete with an average duration of approximately 22 business days per iteration—so 1 month, essentially.

The average number of full time (IT) staff involved in each project was 14.8 and 56 % of projects comprised 50,000 LOC or less, which we would characterize as small to medium in scale.

Clearly, then, the survey drew responses from a pool diverse in experience, responsibility, domain, and scale, which is what we had hoped. To view the complete survey results on participant and project characteristics in a graphical format, we suggest to the reader to view these charts at: http://goo.gl/KJ3l2O

Comparing the population of respondents in this survey to the ones from the past two surveys respondents of the three surveys came from very similar work environments in terms of industry, number of staff, and annual budget. There was also a similar distribution with respect to the application domain and project's duration and length. The respondents in this survey more frequently identified themselves as programmers/developers, software/system engineers or testers than project/product managers than in the past two surveys. While the past two surveys mainly drew participants from the United States, this survey drew participants worldwide.

4 Software development practices

We present the main survey results in graphical format for brevity. We discuss key trends and noteworthy items throughout, leaving other observations for the conclusions.

Perhaps the most interesting finding was that the reported usage of Agile development methodologies (e.g. SCRUM, Extreme Programming, Feature Driven Development, Lean) is almost double that from the 2008 survey—overall, 46 % of respondents indicated that they used an Agile methodology as a software development lifecycle (SDLC) making Agile the most popular SDLC (Fig. 1). This constitutes a significant, if not surprising shift. In both prior surveys the Waterfall model still dominated despite the myth at the time that its demise was imminent [6].

Further analysis linked this question's response to both the respondent's geographic region and to the type of industry. We observed that Agile methods were more than seven times more popular than Waterfall outside the Unites States and two times more popular within the Unites States. Indeed, Agile methods are more prevalent than Waterfall in every region in the United States except of the "Northeast", where they are equally employed.

Looking at the industries employing different lifecycle methodologies we found that Agile methods again outstrip the Waterfall model in almost every category barring the finance/banking/insurance industry.

In light of the changing lifecycle landscape, it is interesting to look at software development practices closely associated with specific development methodologies, and to see if those are changing concomitantly. One such practice is prototyping. In the previous surveys it was always surprising to find that prototyping was heavily used despite the reported popularity of the Waterfall, which traditionally does

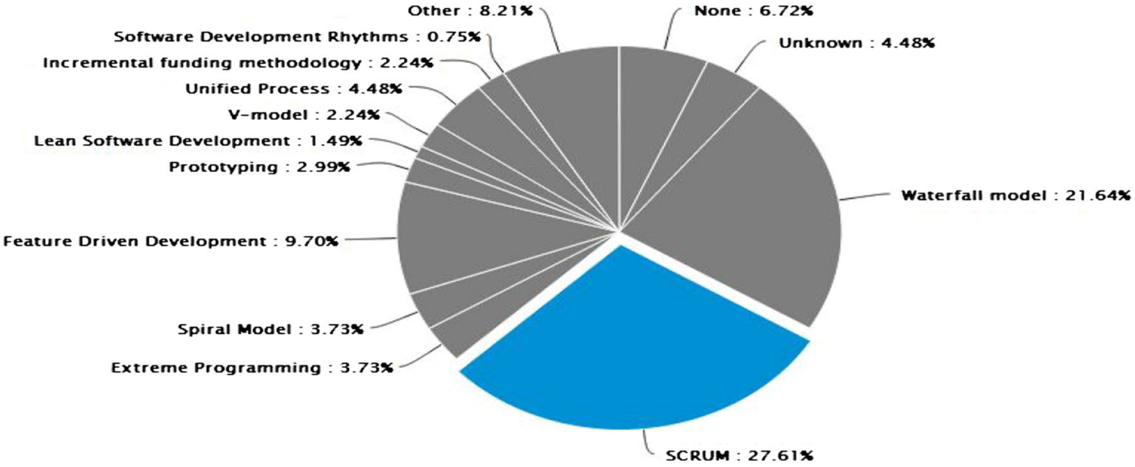
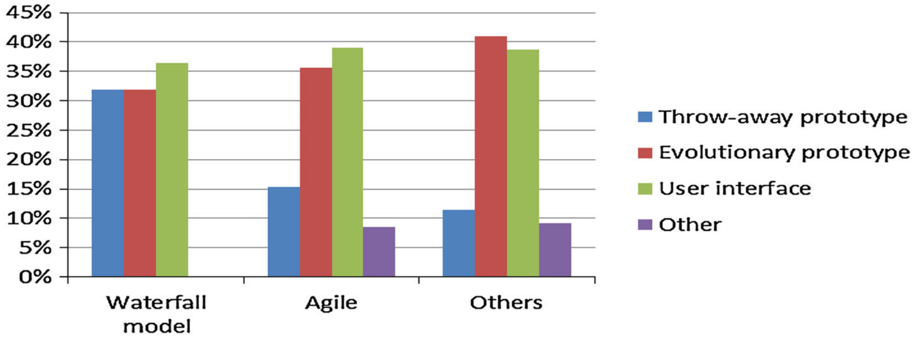


Fig. 1 Software Development Lifecycle employed, n= 134 (Question 9)

Fig. 2 Prototype methods selection across SDLC methodologies (n = 125)



not include their use. So it is surprising, again, to discover that even though Agile methods often include prototyping, their reported use actually declined (albeit very slightly) since 2008, from 75 to 72 %. Furthermore, the use of prototyping was generally no more common in Agile projects as it was in any other, including Waterfall (see Fig. 2). The most marked difference is telling, however. The use of throw-away prototypes is sharply different with waterfall projects, likely reflecting the emergence of refactoring as a practice of design repair in Agile projects negating the need to throw-away hastily developed code.

The survey then focused on techniques used for requirements elicitation, representations, and modeling. We presented an extensive list of known techniques; participants could select all that applied. The data (see Fig. 3a) revealed that 65 % surveyed used "Brainstorming" in the requirements phase. This was the most common approach when any approach was used. While in the 2002 and 2008 surveys, "Scenarios" was the most popular method employed (at 50 % in 2002); we observe that the usage of "Scenarios" is declining from the last two surveys (34 % reported its usage in this survey; dropping down to the sixth position overall). In Agile projects "Scenarios" was less used than in the Waterfall projects. Scenarios were ranked the third most used technique in Waterfall projects while it ranked seventh in the Agile projects. The dominance of Agile methodological use reported likely explains the decline of the "Scenarios" technique. Contrast this difference with the fact that a later question revealed that 38 % of the survey population reported using object-oriented analysis—a technique often applied along with use scenarios. With the decline of "Scenarios" we expected a lower proportion using OO analysis techniques compared to the last surveys (30 % in 2002).

Furthermore, given the dominance of Agile among the respondents in this survey and remembering that Agile practice suggests not to spend much time on documenting requirements, it was interesting to see a rise in the proportion of the population that reported any use of requirements analysis and modeling techniques compared to the previous surveys (54 % in this survey compared to 45 % in 2008) (see Fig. 3b). This is especially noteworthy when we isolate perceptions of quality from those who used analysis and modeling techniques from those who did not. Of those who did not use any modeling methodology, 69 % felt that the finished product's capabilities fit well the customers' needs and only 48.5 % felt that end users found finished products easy to use. Contrast this with those who did employ a modeling methodology where 87 % felt customer needs were met and 82 % believed their

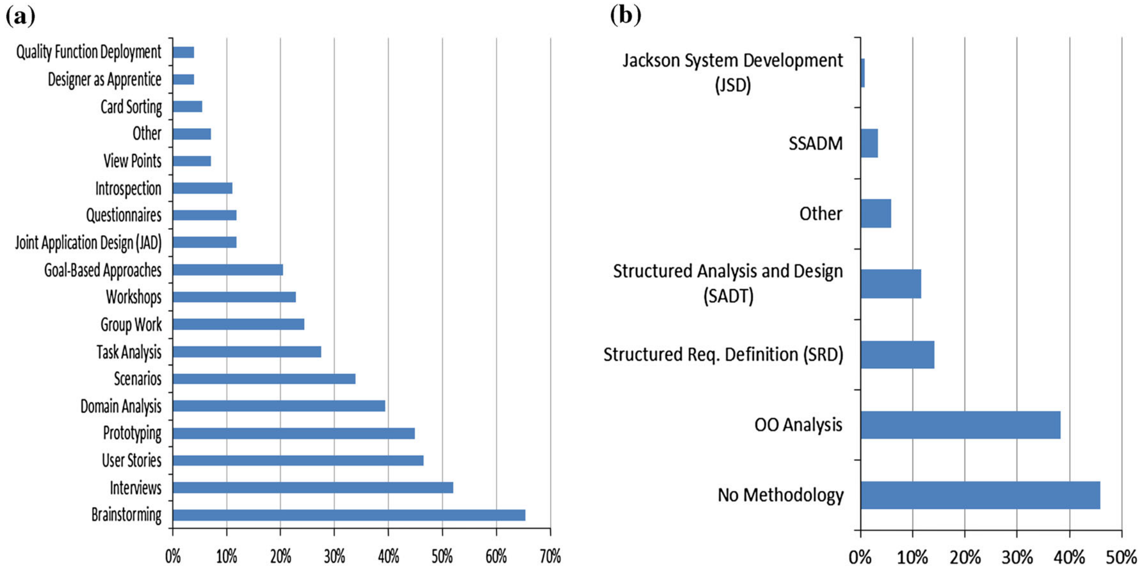


Fig. 3 Techniques used for requirements: **a** elicitation and **b** modeling

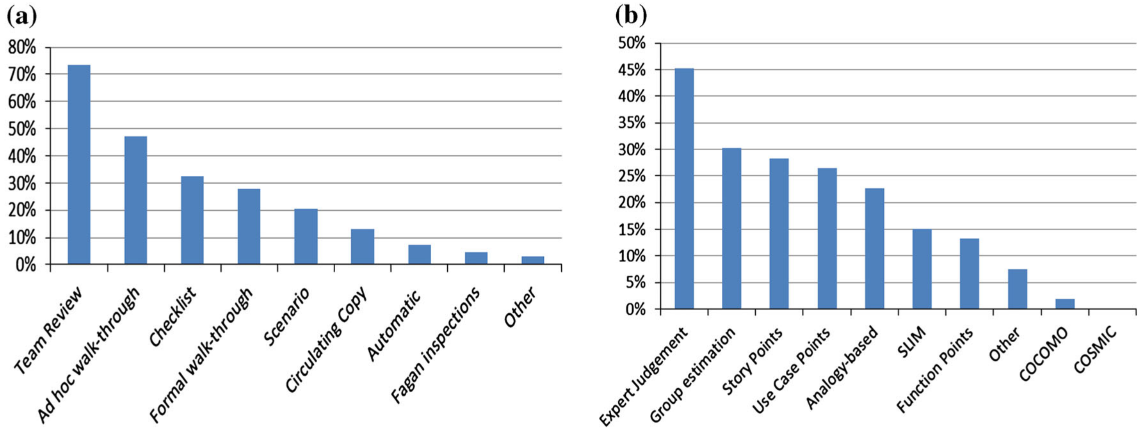


Fig. 4 Requirements: **a** inspection techniques, **b** size/effort estimation

product was easy to use. Clearly, the perception is that these techniques have merit.

The next few questions in the survey focused on the requirements representation's degree of formality and how those requirements were managed. The majority of users (61 %) still report that requirements are expressed in terms of natural language. This number represents an increase from the 2008 survey where 53 % reported expressing their requirements in natural language. We see that only 33 % of users reported utilizing semi-formal notations such as UML in this survey. This is close to its level usage from the last survey (30 %) and it is also consistent with the number of respondents who reported using OO analysis techniques in Question 14. It is also of interest that formal specifications techniques are still not commonly utilized (<1 %). We then wondered whether this lack of formality impacted end-product quality. To answer this, we compared the responses to two questions regarding suitability and usability ("End users found the finished product was easy to use" and "Capabilities of the finished product fit well with customer or user needs"). 86 % of respondents who used semi-formal representations agreed with the first statement—compared to 59 % indicated informal. We saw similar results for the second statement: 90 % who reported semi-formal agreed with the statement, 65 % informal.

An interesting corollary to the formality discussion is the subsequent requirements review and inspection effort. We found that 55 % of respondents are performing inspections. This number is close to 2008 survey result (53 %). That group of respondents apparently employed a range of techniques with an average of 2.29 techniques per respondent (see Fig. 4a).

Out of the 60 % of those who reported on performing estimation for the size of requirements or the effort of building

Table 2 Level of satisfaction (agreement and neutral) with regard to RE practices applied in the project and the company

	Level of satisfaction (agreement and neutral) with regard to RE practices applied in the Project		Level of satisfaction (agreement and neutral) with regard to RE practices applied in the Company	
RE practice	Satisfaction or neutral (2013)	Satisfaction or neutral (2008)	Satisfaction or neutral (2013)	Satisfaction or neutral (2008)
Software quality management	75 %	57 %	68 %	60%
Software development lifecycle	80 %	69 %	74 %	60%
Requirements analysis and modeling	79 %	Not asked	69 %	Not asked
Requirements validation	77 %	Not asked	64 %	Not asked
Requirements traceability	76 %	Not asked	76 %	Not asked
Estimate the size/effort	75 %	Not asked	67 %	Not asked
RE in general	82 %	67 %	63 %	67 %

them, only a minimal proportion reported on using any formal size/effort method (e.g. COCOMO, COSMIC, function points) (see Fig. 4b). Out of this group of respondents, 62 % reported on taking into account the Non-Functional Requirements during the size/effort estimation.

5 Software quality and productivity

Respondents were asked a series of questions meant to assess the level of quality and productivity the project achieved. Overall, respondents agreed that the quality of their software was good and that it met end user needs. 74 % of respondents agreed that capabilities of the finished product fitted well with customer or user needs. Error severity was not significant in the project most of the time with 79 % of the respondents reported either an agreement or neutral responses to that. Also 68 % of the respondents agreed that end users found the finished product was easy to use.

However, questions relating to delivery timeline, schedule and costs indicate that the projects represented in this study took longer than the respondents had expected to deliver. Only 48 % of the respondents agreed that the duration of the project was within schedule; and only 21 % agreed that the project goals were achieved earlier than predicted. Also only 45 % agreed that project costs were within budget estimates.

When they were asked whether corrective hours resolving run-time problems were minimal, 49 % of the respondents had an agreement. On the other hand, 45 % of the respondents agreed that the project could have been completed faster, but that would have meant a lower quality product.

With respect to the development team's quality, overall the respondents agreed that the quality of the development team was acceptable for the project with 79 % of the respondents reported agreement responses. Also, 71 % agreed that the ability and previous experience of the software development team was adequate and 69 % agreed that the team size was adequate for the project.

With respect to their level of satisfaction with regard to the efforts of requirements engineering practices applied in their project and to their company in general respondents reported that they were either satisfied or neutral in their level of satisfaction with these efforts according to Table 2 (we also included the corresponding results from the 2008 survey for comparisons).

6 Conclusions

Throughout this paper we have reported on the changing landscape of requirements elicitation, analysis, modeling, and verification with respect to reported practice over the last decade. The comparison of surveys has revealed some interesting trends including the continued rise of Agile methods and their accompanying practices. More revealing, however, is that some techniques and paradigms have still not risen to dominance as we would expect. Object-oriented analysis and design and the UML, for example, are reportedly far less common than we anticipated, which could be considered disappointing given the improvements in end-product capability and ease of use that such techniques reportedly provide. Some other salient observations that emerge from this survey and those prior are:

- A number of requirements in engineering practices show no significant changes from the 2008 survey including SQM, Requirements Inspection, and prototyping—although throwaway prototyping has seen a significant decline with the emergence of refactoring.
- Overall, respondents expressed much greater satisfaction with the use of RE practices than was reported previously.

Our hope is that researchers will use this data to corroborate their own research and to motivate follow-up research studies. Our own subsequent work will offer more detailed analysis of some of the 2013 results.

References

1. Neill Colin J, Laplante Phillip A (2003) Requirements engineering: the state of the practice. Software 20(6):40–46
2. Marinelli V (2008) An analysis of current trends in requirements engineering practice. In: Master of software engineering professional paper, Penn State University, Great Valley School of Graduate Professional Studies, Malvern
3. Khurum M, Uppalapati N, Veeramachaneni RC (2012) Software requirements triage and selection: state-of-the-art and state-of-practice. In: Proceeding of the Software Engineering Conference (APSEC), 2012 19th Asia-Pacific, vol 1, pp 416, 421, 4–7 Dec 2012
4. Chernak Y (2012) Requirements reuse: the state of the practice. In: Proceeding of the IEEE international conference on software science, technology and engineering (SWSTE), June 12–13, 2012, p 46, 53
5. Verner J, Cox K, Bleistein S, Cerpa N (2007) Requirements engineering and software project success: an industrial survey in Australia and the US. Australas J Inform Syst 13(1)
6. Laplante PA, Neill CJ (2004) "The demise of the waterfall model is imminent" and other urban myths of software engineering. ACM Queue 1(10):10–15