Marginal Notes on Amethodical Requirements Engineering: What experts learned from experience

Susan Elliott Sim, Thomas A. Alspaugh, and Ban Al-Ani
Department of Informatics
University of California, Irvine
{ses,alspaugh,balani }@ics.uci.edu

Abstract

Requirements engineers with many years of experience have a distinct perspective on the field. To sample this knowledge, we interviewed 34 requirements researchers and practitioners, each with up to 42 years of experience. We used open-ended, structured interviews in which we asked them to reflect on their experiences and professional development as requirements engineers over their careers. Several themes emerged: requirements engineers act as bridges between different worlds, good communication is key, good process can help but isn't everything, shorter requirements documents can be better, and good requirements are driven by customer value not technical elegance. All of these pertain to amethodical requirements engineering. *Amethodical concepts are not rejections of method, but rather those concepts that are marginalized and left out of prescriptive methods for carrying out a procedure. We discuss these results and their implications.*

1. Introduction

There are some things you can only learn from experience, and these are the kinds of knowledge we sought to investigate in our preliminary study of requirements engineering expertise. We wanted to characterize the differences between novice and expert requirements practitioners. In a review of the literature, we found taxonomies of skills, knowledge, and activities that requirements engineers should master. But before proceeding to hypothesis-driven research, we were interested in exploring whether these existing taxonomies were complete and sufficient. To this end, we undertook a qualitative study of experts in requirements engineering. We used a "war stories" approach and asked participants what they had learned through experience.

As the study proceeded, we were surprised and intrigued by what our subjects told us. Contrary to our expectations, no one in the study had the job title of "requirements engineer," none of them had been permitted to do requirements engineering until they had acquired experience in software development or in a problem domain, and no method, tool, or notation was mentioned by more than one participant as essential knowledge. However, we found common themes in their reflections. The participants spoke of the need to act as bridges between groups that normally have no interaction. They emphasized the importance of using language well and diplomatically to communicate with customers and write requirements that were easy to understand. They felt that having a good requirements process could be helpful, but being flexible in its application was more important. They had seen first-hand that in requirements specifications, less is often more. Finally, they believed that requirements should be driven by the business case rather than by technical elegance.

In analyzing the participants' responses, we were struck by how rarely they mentioned the material that is typically considered the core of requirements engineering, such as tools, techniques, processes, and notations [17]. These comments present a different figuration of requirements engineering from that which is normally seen in textbooks: *Figuration* is an analytic technique for clarifying subject-object distinctions and has been applied to topics such as history of science and technology [11]. By placing a concept in the subject position, or foreground, other concepts are necessarily put in the object position, or background. Using this lens allows one to select which concepts are foregrounded, thereby attaining a more inclusive view a phenomenon. We apply figuration to the methodical/amethodical distinction developed by Truex, Baskerville, and Travis [23]. They

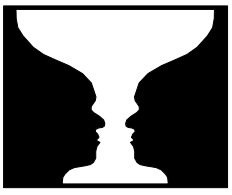

Figure 1. Goblet or faces?

apply postmodern deconstructionism to information systems development texts in order to identify a *privileged* interpretation in a reading of a text, i.e. the assumptions and ideals underlying the central message. For every privileged text, there is a *marginalized* interpretation, i.e. a ratrionset of assumptions and ideals that are deferred as a result of being in the object position. According to Truex et al., the privileged interpretation in information systems development is *methodical*, that is, a view of the world as ordered, rational, and logical. The marginalized interpretation is termed *amethodical*, that is, a view of the world as capricious, random, and socially constructed. To be clear, amethodical does not mean careless or without a procedure, but rather, beside or outside of method.

Participants in our study typically made comments that fit an amethodical world view. *"Every project is different so don't be rigid. Be observant and listen,"* stated Scott, a business analyst with fifteen years of RE experience (all participant names are pseudonyms). James, a business architect with over twenty years of experience, noted, *"Context. Everything is contextual. Everything."*

Truex et al. do not try to argue that methodical is better or more prevalent than amethodical, and neither do we. They simply use the concepts to draw attention to the dichotomy, as we do. Every development project displays both methodical and amethodical aspects. Similar observations have been made by other researchers in requirements engineering. For example, Potts has identified two philosophical stances in requirements, which he termed "abstractionism" and "contextualism" [18]. The former aligns with, but is not identical to, the methodical view, and the the latter aligns with the amethodical view. Depending on which view or philosophy is emphasized, different knowledge, skills, and abilities are brought to the foreground. Selecting a figuration has implications for how we view requirements engineering knowledge, how we measure expertise, what practical problems we see, what approaches we apply to solve a particular problem, what research questions are valued, and how we classify and evaluate research work.

The remainder of the paper is organized as follows. Related work is discussed in Section 2. We describe our study design over two sections, Sections 3 and Section 4. In Section 5, we discuss the participants in the study. We summarize our data and present examples in Section 6. Section 8 addresses threats to validity. We discuss implications of our analysis in Section 7, and Section 9 concludes the paper.

2. Related work

There are two areas of related work for this research. We consulted the literature on expertise and requirements expertise in the design of our study and formulation of our research questions. In addition, we examined prior work on amethodical systems development and the nature of requirements engineering as we proceeded through data analysis.

2.1. Expertise

Expertise has been defined as consistently high performance on a set of tasks in some area of human activity [22]. It is usually accompanied by a record of achievement that cannot be explained by chance alone. Psychologists have studied expertise in a number of fields, including chess, mathematics, physics, sports, and music, to better understand cognitive structures and to improve pedagogy and training [8]. The most widely cited result from these studies is that it take 10 000–20 000 hours of deliberate practice, or a minimum of 10 years to become an expert [8].

While the novice-expert distinction is well understood as important, there has been little work on expertise in requirements engineering. Sutcliffe and Maiden in 1992 who studied novices problem solving as they wrote a requirements specification [21]. As well, Hickey and Davis studied the elicitation methods used by requirements experts using a method similar to ours [12]. However, there have been many empirical studies that have categorized subjects as novices or experts. In addition, other writings mention the importance of skills and expertise in requirements [10, 25].

We began our study of expertise in requirements engineering by looking at categorizations or taxonomies of skills, knowledge, or activities, such as those found textbooks and survey papers [17]. Using the literature and findings from psychology, we developed an initial model for requirements expertise that characterized different levels of ability, such as naive, beginner, novice, intermediate, expert, in the usual skill areas, such as requirements elicitation, analysis, communication, validation, and management [1]. We wished to assess the completeness of our model, so we undertook an exploratory study of expertise.

2.2. Figuration

As we analyzed our data, the concepts of *figuration* and *amethodical systems development* rose to prominence.

A famous Gestalt visual illusion looks like either a goblet or two faces in profile, depending on how one focuses one's attention, as shown in Figure 1. This process of choosing an image to bring to the foreground is called *figuration*. Bringing one image to the fore necessarily pushes the other into the background. It is not possible to see both at once, yet the image has not changed at all, only one's perception of it.

Haraway uses the concept of figuration to examine the bi-directional process of scientists defining the natural world [11]. By bringing the scientist to the foreground, the natural world becomes the created object. By bringing the

natural world to the foreground, the scientist becomes the created object. This same approach was used by Truex, Baskerville, and Travis [23] who used postmodern deconstructionism to analyze texts prescribing procedures for developing information systems. Truex et al.'s analysis is an example of figuration. By analyzing these texts, they pull amethodical systems development to the fore. Truex et al. do not try to argue that the methodical figuration is better or more prevalent than amethodical; they simply use the concepts to draw attention to the alternative views. They also advocate being aware of both views and taking them into consideration when developing systems and when doing research into systems development. All the themes that emerged in our analysis of the data are amethodical and represent a possible figuration for requirements engineering.

3. Research questions — what experts know

The goal of this initial study was to validate our preliminary taxonomy of requirements engineering expertise. We wanted to find out whether our taxonomy reflected the actual knowledge, skills, and traits found in practice. To this end, we undertook an exploratory study using interviews of expert requirements engineers.

Our research questions were:

- Did our model of skill areas and levels accurately reflect requirements engineering expertise?
- What was missing from our model?
- What kinds of things did requirements engineers learn through experience?
- How did people develop as they became expert requirements engineers?
- What were the kinds of mistakes that people make along the way?

While our initial intention was to answer the first question, it quickly became clear that our model was missing a great deal. Consequently, our analysis shifted to the last three questions. The answers we found are the focus of this paper.

4. Method

We undertook a two-stage qualitative study [14] to address the research questions.

In the first stage of the study, we sought participants at the 2006 International Requirements Engineering Conference. They filled out a questionnaire with demographic data, primarily background and experience. The goal of this stage was to do an initial assessment of the distribution of experience the population of requirements engineers, and guide selection of participants for the second stage.

In the second stage of the study, we interviewed participants to gather data relevant to the research questions. The 34 interview participants fell into three groups:

- 14 selected participants from the RE'06 questionnaire,
- 15 practitioners at Intuit, Inc. [13] in San Diego, and
- 5 practitioners elsewhere in industry in Southern California.

All the participants from RE'06 were highly experienced, and could easily be considered the elite of the field. To complement this highly qualified group, we sought additional participants with less experience. We specifically sought novice requirements engineers as participants from Intuit, and found none. The further five participants were sought as an indication of whether this experience distribution was general throughout industry. As we will discuss in the next section, there were no meaningful differences in the responses from the groups. We heard many of the same sentiments equally from the industry practitioners and the RE'06 group.

1. What do you think a novice requirements engineer should be able to do?
2. What do you think an expert requirements engineer should be able to do?
3. Please rate your level of expertise.
4. Can you compare what you do now to what you did when you first started out as a requirements engineer?
5. (a) *(If interviewee is an expert requirements engineer)* What advice would you give someone on how to become a better requirements engineer?
   (b) *(If interviewee is a novice requirements engineer)* What do you think you would need to learn to become a better requirements engineer?
6. Tell me about a time when involving an expert requirements engineer in a project was advantageous.
7. Tell me about a time when involving a novice requirements engineer in a project was detrimental.
8. Is there anything else you would like to share? Is there a question that you think I should have asked?

Figure 2. Interview script

Each interview lasted approximately thirty minutes, and consisted of open-ended questions from a script (Figure 2) and follow-on questions for further exploration. Questions 4 and 5 are based on questions used by Brown, Campbell, and DiBello [5] who studied how programmers proceeded through an intellectual developmental sequence as they acquired expertise. Questions 6 and 7 employed the war stories technique pioneered by Lutters and Seaman [15] to probe a phenomenon in context. Interviews were audio-recorded and transcribed. The transcripts were analyzed by chunking and open-coding [7, 14].

5. Participants in the study

Our first set of findings concerns the characteristics of the participants in the study. Table 1 summarizes the experience and background of the study participants. Each subject has been given a pseudonym to protect his or her confidentiality. All of them have a great deal of experience, with a median of 20 years of industry experience (interquartile range = 10.5 years), with a median of 15 years of experience working with requirements (interquartile range = 10.75 years). Our participants had worked on a variety of projects including: information systems, financial software, transportation, aerospace, vending machines, aircraft, and naval ships. They had a total of 46 requirements-related publications among them, in venues such as IEEE Requirements Engineering Conference, Requirements Engineering Journal, International Conference on Software Engineering, and IEEE Transactions on Software Engineering and Methodology. By all measures, this group was highly experienced in requirements engineering.

5.1. No "Requirements Engineers"

Although everyone interviewed worked with requirements in one form or another, none of them had the job title of "Requirements Engineer." Common job titles were manager, consultant, system engineer, software engineer, business architect, and business analyst. As well, many of the participants struggled with the question about what they expected novice or expert requirements engineers to be able to do, both due to the categorization of skill levels and the idea of requirements as a separate discipline. They often turned the question back to us for clarification, or re-stated the question in a form that they could answer. Aaron took issue with the job title, *"Is 'requirements engineer' a specific job or is it a capability that people have developed while doing other related jobs? I have never met anyone holding an exclusive title of 'requirements engineer.''* While Irwin questioned the label for the discipline, saying, *"I think it would be difficult to talk about requirements engineering without skill in politics, skill in facilitation, managing people and some sort of female skills such as intuition and sensitively assessing situations. These don't feel very much like engineering to me. I think the name is amazingly flaky. It is a very peculiar name. ...Because requirements are human needs aren't they? Are requirements more like therapy? Is it engineering at all?"*

Despite the absence of this job title, we will continue to use the term "requirements engineer" in this paper. It is a serviceable generic term for all of this job titles and it describes someone engaging in the collection of activities of interest to the requirements engineering research area.

5.2. No Novice Requirements Engineers

Within our group, there were no novice requirements engineers. We were not surprised by this paucity following the interviews at RE'06. However, when we conducted interviews with participants from industry, we made a special effort to find them, but were not successful.

Among the professionals, our participants tended to have a few years of work experience before focusing on requirements-related work. A few exceptions are managers, such as Anita, Bruce, Jane, and Mark, who supervised people who worked with requirements, but didn't necessarily do requirements work themselves. Generally, people are not permitted to do requirements work on their own immediately after graduating from school. Often, a novice is given simple tasks and is paired up with a senior requirements engineer, so he or she can be mentored or supervised. Irwin is one person who believes in sheltering inexperienced employees, *"The requirements engineer who has no strong deep work experience— I would expect him to observe and add value without being put in a hazardous position where they can either sink the project by making mistakes out of domain ignorance or put themselves in personal peril by being so wrong that they blow all credibility out of the water."*

This observation is consistent with Aranda, Easterbrook, and Wilson's [2] finding that the CEO of a small company retains responsibility for requirements engineering long after other technical activities have been passed on to others. Doing requirements well requires experience and seniority and is too important to the life of the organization to entrust to just anyone.

5.3. No Single Path to Expertise

Our participants came to requirements engineering from a variety of backgrounds and acquired their expertise in a variety of ways. While many of them felt it was necessary to maintain their knowledge and skills by reading books, keeping abreast of current technology, taking courses and tutorials, and attending conferences. There was a great deal of variability in how they evolved as requirements engineers. To give a flavor of this diversity, profiles of four of our participants are given here. They are taken from across the three groups and include a software engineer, a consultant, an aerospace engineer, and a business architect.

Derek is a software engineer with an MS and MA and a CS/IS and Math background. He is currently employed as a software and requirements engineering consultant. He has 30 years industry work experience and 20 years in requirements engineering. He advised learning from a good set of books on all the aspects of requirements engineering, bolstered by hands-on experience. He also highlighted the

Code name	Years in industry	Years in RE	Industry positions	Academic positions	Education	Pubs
RE'06						
Iona	42	>20	Consultant		Math related MS	2
Norah	40	15	Consultant		Math related PhD	6
Mark	35	20	Software System Engineer		Aerospace related PhD	2
Derek	30	20	Software Engineer, Consultant		CS/IS related MS, Math related MA	0
Jacob	25	10	Consultant, Systems Engineer		CS/IS related PhD	3
Raymond	23	7	Software Engineer, Consultant, Manager		CS/IS related MS, MBA	0
Irwin	>20	>10	Software Engineer, Consultant, Systems Engineer		CS/IS related MS	9
Edwin	19	15	RE Change Agent		Math related MS	3
Anthony	17	29	Software Engineer, Consultant, Manager	Lecturer, Postdoctoral Researcher, Professor	CS/IS related PhD	2
Aaron	15	10		Professor	CS/IS, Business related PhD	3
Morgan	10	7	Software Engineer, Consultant, Manager	Postdoctoral Researcher	CS/IS related PhD	0
Muriel	9	1	Consultant, Project Manager		Marketing/Business related BS	0
Craig	>20	20		Professor	Psychology related PhD	8
Stephen	7	22	Consultant, Manager	Professor	CS/IS related PhD	8
Intuit						
Erica	30+	10	Manager		Math related MS	0
James	26	23	Business Architect		CS/IS related Certificate Program	0
Malcolm	25	20	Software Architect		CS/IS related BS	0
Scott	25	15	Business Analyst		CS/IS related BS	0
Dan	21	21	Software Architect		CS/IS related BS	0
Myron	21	15	Business Architect		Telecommunications related MS	0
Howard	20+	5+	Software Engineer, Manager		CS/IS related BS	0
Anita	20	20	Manager		Economics and Management Science related BS	0
Bob	20	12	Software Engineer		CS/IS related BS	0
Bruce	19	19	Manager		CS/IS related BS	0
Joann	18	10	Manager		CS/IS related MS	0
Tracy	18	6	Software Process Improvement Manager		Cybernetics related BS	0
Audrey	13	7	Software Engineer, Manager		CS/IS related BS	0
Jane	10	10	Software Designer		Industrial Engineering related BS	0
Carol	10	2	Software Engineer		EE related MS	0
Other industry						
Mike	17	4	VP of Product Management		EE related BS	0
Kevin	20	10	Director of Product Management		EE related BS	0
Ray	18	4	Senior Globalization QA Manager		CS/IS MS	0
Mark	25	25	Product Manager		CS/IS MS	0
Roger	28	18	Director of QA		CS/IS BS	0

Table 1. Overview of study participants

importance of social skills, with an emphasis on interacting with the real customers.

Norah is a consultant with a PhD in Math and has a total of 40 years experience in industry, 15 of which has been in requirements engineering. In her interview, she emphasized the importance of social skills in carrying out collaborative activities. She stressed that requirements engineers should continuously update their knowledge of technologies and techniques.

Mark has a PhD and 35 years in the software systems industry, with 20 years in RE, and is currently employed in the aerospace industry. Mark also emphasized the need to maintain awareness of where to find information to support the learning process and to keep up with current trends in requirements engineering. He actually lowered his personal assessment of his expertise, because he has not kept up with the technologies available.

James is currently employed as a business architect and has a certificate program degree in CS/IS. He has 26 years experience in industry and 23 years experience in requirements engineering. He was the only participant in the study who rated himself an expert. His main emphasis in the interviews was taking the realities of life into consideration and contextualizing information within the big picture.

6. Results and observations

In the previous section, we discussed commonalities we observed among our diverse group of participants. In this section, we discuss the commonalities among their anecdotes and reflections. Paramount among these is the idea that a requirements engineer is someone who builds connections by becoming a bridge spanning business and technical considerations, the range of stakeholders, and the problem domain and system, through sensitive and effective communication. Other themes that emerged were: natural language skills and good communication are key; good processes can help, but need to be used selectively; writing less can result in better requirements; and business value must be the primary consideration in requirements.

6.1. Requirements engineers are bridges

Requirements engineers are bridges between people who don't normally interact, between the business and technical worlds, and between the problem and the solution. Because they need to span different worlds, they need to rely heavily on skills such as communication, diplomacy, negotiation, organizer, and leadership. James illustrates this: *"I created this whole requirements document that was really business requirements and gave it to this team and they said, 'But what do I do?' (laugh) And I had to figure out how to take it to a level that they could work from. That wasn't exactly the easiest job, because I thought I had done a bang up job on the requirements. ...I failed to understand my audience well enough to give them something they can work from. However, I still need to be accountable in my role back up into the business, and so I provide this bridge back and forth and I need to be cognizant of both sides of the bridge."*

Raymond believes pulling together diverse people is key to producing good requirements: *"This notion of a requirements team is, I think, is really important with cross-functional requirements teams with different functional areas that are committed to working together on developing requirements, implementing the requirements from their different perspectives. ...development engineering, marketing (people who talk to the customers), verification and validation (to make sure that what you're doing is testable), and sometimes finance or business unit folks (in order to evaluate ROI, we do this versus that, tradeoff decisions)"*

Sensitive and effective communcation can facilitate building bridges. Anthony witnessed this practice in his mentor: *"I learned the business of dealing with customers through [the company president]. I basically shadowed him. ...And I was able to see how* tender *he was. Here's a PhD in mathematics, who was* tender *with his customers. He* cared *about his customers. And that told me that, ooooh, being a PhD doesn't mean that you have to be a hard ass. You can be a* soft *person and understand and listen. ... That taught me a lot. You won't learn that in engineering."*

Irwin described a novice requirements engineer failing to be a bridge: *"When I was just starting out, I did some interviews with a media company and I think there was one guy who was explaining something about himself. He was explaining something about a part of the technical domain. And I did something fairly chin first like 'Ahh so that means this category as opposed to that category.' And he basically violently disagreed with what was more or less self evident from what he had said the sentence before. And people can get cross at you asserting something. Whereas, if you just delicately say, 'Does that mean that?' or 'Could you explain that?' By elegantly going around things it is possible to avoid getting any kind of stressed out reaction from people. People can be very sensitive."*

6.2. Natural language skills are key

Participants emphasized that the most important requirements engineering language is natural language, and that natural language skills, both verbal and written, are key to achieving good requirements. Verbal skill does not mean good grammar and diction in this context; it means communicating effectively with the listener. Several participants spoke in an earthy, unrefined manner with many self-interruptions, but were easy to understand and disarmingly personable. Malcolm, a business architect from Intuit, was

seemingly sloppy in his speech, but clearly expressed the importance of communication. He spoke for many of the particpants when he said *"Just be clear. Have a good command of the language you're delivering the requirements in. ... written and verbal communication are ... that's just key. If you don't have a command of how to put a requirement on paper that someone can understand. Sitting there and talking about it and whiteboarding about it all day will do you no good. You need to be able to communicate effectively. ... You need to be able to take as much of the ambiguity out as possible. In some cases, you can't, but you need to be conscious of that, all the time. Back and forth, getting rid of ambiguity, and making sure that you understand both sides. You understand the business and you understand the engineering. It's a tough job to do well."*

6.3. Good process isn't everything

Proven processes, best practices, and tested solutions can help requirements engineers avoid unnecessary work and known pitfalls. However, they need to be used selectively. Audrey illustrated the benefits of good process through an anecdote in which her team failed to follow one. An internal customer asked them to implement a framework in another programming language "just like" an existing framework in Java. *"We thought we understood it and we documented it, but we never really went back and confirmed it with them or the magnitude of it."* They had half a dozen meetings with the customer, but it wasn't until Audrey's team was training them to use the resulting product that anyone realized it wasn't what they wanted. She said, *"The good thing is it was caught in time, but the bad thing is we could have avoided it by having follow-up."*

Edwin spoke of why process should be used selectively to achieve specific goals, never for its own sake: *"We've had several people try to do this CMMI or CMM at the company and they are usually enthusiastic but dogmatic people. Eventually there are these organizational antibodies that activate. When you're seen as serving a process-related goal as opposed to a business-related goal you get ejected by the organizational antibodies. They mobilize, eat you, and then you're gone. You're a contagion that's taken out."*

6.4. Less can be more

Longer requirements documents are not necessarily better. Often, writing requirements at an appropriate level of abstraction is more important than including all the details.

Edwin felt novices often *"take the easy path instead of the correct path where they write the easiest 50 pages of requirements they can think of to feel that they are productive rather than the most valuable five pages that would have been much better even though they are more contentious or more difficult. Fine, but they're more beneficial."*

Stephen spoke of the requirements for a new version of an existing system. *"This document kept growing in size as there were around 150 requirements and each requirement took some pages."* His solution was to create two smaller documents, one the "conceptual solution" with the business case and the other the "version definition" containing the details. *"The version definition, therefore, was much thinner and was used as a decision-making document, whereas the conceptual solution document was used for determining the content of the software."* By using smaller, more concise documents, both the business stakeholders and the software developers were able to make better decisions.

Iona summed up this lesson of quality over quantity as, *"An expert should also be able to recognize if the requirements are at the right level of abstraction and detail."*

6.5. Business value drives requirements

The final lesson in this section is that there always needs to be a business reason behind requirements. Aaron, a professor with ten years of industry experience, recalled being a novice who had lost sight of the business value behind a product. *"We developed quite a few products in the past which had no market at all, because we were implementing nice ideas that we had as engineers, or were technically advanced."*

Bob has taken this lesson to heart when he works on requirements. *"When I just started out I was interested in getting to the solution quickly and now I'm interested in getting to what the solution should do for the business. That's a huge difference. Once you have that, then everything else after that falls into place."*

7. Discussion

We noted in Section 2 that all five themes we identified are predominantly amethodical.

Requirements engineers are bridges between worlds The worlds to be bridged usually vary from project to project, as do the ways in which they can be bridged. The idea that worlds need to be bridged is inherently amethodical, because in the methodical view, multiple spheres don't exist and a logical system doesn't need bridging.

Good communication is key Good communication appears to be largely amethodical, in the sense that every good communication has "a unique and idiographic form" matched to the situation, the participants, and the matter to be communicated.

Good processes help, when used selectively Processes are methodical, by definition, but the selective use of process as described by the participants seems to be the result of

experience and judgement. The participants also note that a good process is not sufficient for a good result, and the additional desiderata are amethodical.

With the appropriate abstraction, less is more The choice of the appropriate abstraction is aided by methodical knowledge and techniques, but again the methodical does not suffice. Producing a requirements document with less detail that more effectively meets the needs of everyone involved is consistent with the idealization of change and choice. According to the methodical view, which idealizes generalization and consistency, a single comprehensive document is preferred.

Business value, not technical elegance, should drive requirements Business value is situated and ad hoc, varying from situation to situation. In contrast, technical elegance is based on general principles that are rational and universal. Favoring business value is amethodical, while favoring engineering process is methodical.

In each of these cases, we see that a substantial part of the aspect described is amethodical. As noted by Truex et al., amethodical is a concept that is marginalized in method texts. Books and academic courses tend to emphasize general principles of lasting value, in other words, the methodical. This contrast between things learned through experience versus things learned through other means has been observed by others. In the Nicomachean Ethics [3], Aristotle identified five intellectual virtues. One of these was "phonesis," or knowledge gained through practical experience. Another was "sophia," a form of knowledge that could be gained through the application of reason alone. Similarly, Russell [19] characterized the distinction as "knowledge by acquaintance" versus "knowledge by description." Both philosphers felt that there was a qualitative difference between the two and each provided access to different kinds of truth about the world.

Truex et al.'s analysis parallels Haraway's technique of figuration, in which a marginalized concept is brought to the foreground, not with the intent of replacing a privileged text, but to introduce a new way of understanding that shifts fluidly between alternative figurations. Given methodical and amethodical interpretations, what kind of requirements engineering research emerges with each figuration?

7.1. Figuring RE Research

Within requirements engineering, a methodical figuration brings certain tools and techniques to the foreground. Within this view, contributions that regularize requirements, use formalisms, and employ logical decompositions are valued. For example, formal methods and model checking depend on and exploit the rationality behind software systems. Similarly, techniques for imposing order, such as ontologies and XML, are promoted and valued skills tend to be analytical in nature. The "soft issues" are pushed to the background and marginalized [9, 24].

If we adopt an amethodical figuration, another view of requirements engineering is pushed to the foreground. From a perspective that views the world as negotiated, capricious, fragmented and creative, different tools, notations, and techniques are valued. Some examples of research that fit with this world view are elicitation methods such as contextual inquiry [4] and workshops to increase creativity in requirements [16]. It should be noted that this view does not exclude tools entirely. Approaches such as Chechik et al.'s work on multi-valued logic [6] and Sabetzadeh's work on merging multiple, sometimes inconsistent models [20] would be appropriate.

Each of these figurations makes an aspect more prominent. Both are present at all times, but one can be easier to see than another. By using the approach of figuration, we can hold a more inclusive picture of requirements engineering in our mind's eye.

7.2. Alternative Figurations

Others have also proposed alternative figurations for requirements engineering. Potts [18] had identified two philosophies in requirements that he termed "abstractionism" and "contextualism." Abstractionism is an engineering approach that seeks to prescribe good designs built using established principles. Contextualism is a user-centered approach that seeks to build software that supports current practice by taking the peculiarities of a situation. There are clear similarities between contextualism and the amethodical, and between abstractionism and the methodical.

A characterization of requirements engineering based on action problems and knowledge problems has been proposed[27]. While this figuration bears less resemblance to methodical/amethodical, it presents another perspective. According to Wieringa, "An action problem is a desire to change the world; a knowledge problem is a desire to increase our knowledge about the world." He argues that the practice of requirements engineering is concerned with the construction of domain theories, which is not an action problem at all, and hence is not really engineering. Taking the proposition further, Wieringa makes the claim that the outcome of requirements engineering research can never be a method. Applying this characterization to the results of our study, the comments from our participants emerged from their experience on action problems and not just knowledge problems.

Extending this characterization, Wieringa, Maiden, Mead, and Rolland [26] proposed a classification for requirements engineering research papers. Their classification includes evaluation research, proposal of a solution, validation research, philosophical papers, opinion papers,

and personal experience papers. The goal of this classification was to establish different evaluation criteria for each kind of paper, so that reviewers and publications can be more inclusive. This classification is a sort of figuration for two reasons: certain types of papers are in the foreground, i.e. those discussed in the classification; and it encourages consideration of other points of view. According to the classification, this paper would be categorized as evaluation research, because it investigates a problem in RE practice.

8. Threats to validity

In this section we discuss the internal and external validity of our study, as well as threats to validity.

Internal validity is the soundness of the conceptual relationships within a study. Our goals here were to accurately determine what the participants believe constitutes requirements engineering expertise in the broadest sense possible, what they have learned from their requirements engineering experience, and how they developed as they themselves became more expert. We developed the questionnaire and interview script in order to elicit responses that address the research questions, and transcribed and analyzed the interviews in order to identify the relevant responses while retaining the context and internal relationships among the specifics of each participant's interview [14].

External validity is the degree to which results from the study can be generalized and provide insight. Our results can be generalized to the extent that our findings are representative of requirements engineering in general. In qualitative research, a good criterion for determining whether a sufficient number of participants has been interviewed is "theoretical saturation" [7]. When the same comments are heard repeatedly in interviews, and no new information is being gathered, then enough people have been included in the study. Typically this happens with 10-30 subjects. By either measure, our 34 participants are sufficient for this study. We interviewed a number of highly-experienced, reflective requirements engineers, as indicated by their years of experience and the compellingness of their responses. These results are rendered more persuasive by their coherence with the responses of other participants at all levels of experience.

We identified several possible threats to validity. The questionnaire and interview scripts might not have elicited responses that address the research questions, or might have only elicited a portion of the relevant responses.

The questions in the interview were deliberately open-ended and intended to solicit information about what requirements engineers have learned from their experience. We addressed this threat by careful design of the questionnaire and script, and careful conduct of the interviews [5]. We used a "war stories" approach and open ended questions to give our participants a great deal of latitude in the answers they provided [15]. The set of participants might not have been representative of requirements engineers as a whole. We addressed this threat by seeking participants from a global pool (attendees at RE'06), and participants specifically from industry (Intuit and other industry locations). Our internal analysis of the data gave no indication that any subgroup of our participants had differing views on the research questions, supporting our goal of a representative sample.

Despite this care, there still exists the possibility of bias in the results. All of the responses from participants in the study were ones that they gave in response to open-ended question that were open to interpretation. From all the different comments that they could have given us, they chose these. The participants were conscious of our roles as researchers, occasionally making comments such as "that would make a good thesis topic for you," "there hasn't been enough research on that," "as a young person you should know ...," or taking issue with our choice of questions or categories that we used. This awareness suggests to us that they framed their answers in the context of a scholarly research project, which could bias the results toward non-academic topics. However, this bias is a possible advantage, because participants tried to tell use about things that we did not know, thereby contributing to the breadth of data that we received.

9. Conclusion

We undertook a study of the kinds of things requirements engineers learn from experience for the purpose of informing out model of requirements expertise. The central themes that emerged pointed to aspects of requirements engineering knowledge that we did not expect. Our analysis of the data highlighted several themes:

- Requirements engineers are bridges between worlds;
- Good communication is key;
- Good processes help, when used selectively;
- With the appropriate abstraction, less is more; and
- Business value, not technical elegance, should drive requirements.

All of these themes are amethodical and are indicative of a world that is random, opportunistic, fragmented, and *ad hoc*. They represent the knowledge that is acquired through experience and is marginalized in texts on systems development. Our participants themselves characterized their own comments as things that you don't learn in school or things that weren't really engineering.

The methodical and amethodical interpretations are opposing figurations of requirements engineering. Like the Gestalt visual illusion of the faces and the goblet, both are

present at the same time, but through a trick of the mind one can flip back and forth between one being in the foreground and the other in the background, or in the subject or object positions. Requirements engineering is the same in that both the methodical and amethodical are present at all time, but depending on which is in the privileged position or marginalized position, certain kinds of ideas become more prominent. Is requirements about the managed and orderly application of technology? Or is it about negotiating through human conflict to arrive at a compromise solution? Each aspect demands a certain kind of research into tools, techniques, methods, and notation. Or knowledge problems and action problems. Both of these figures are in the picture; both are in requirements engineering simultaneously. Only a trick of the mind is needed to bring one or the other to the privileged position.

10. Acknowledgments

Our first debt of gratitude is to the participants in the study who shared their experiences with us. Thanks to Medha Umarji who helped to develop the initial survey instruments. We are grateful to Vivian Olivera and Joey Lei who helped conduct and transcribe interviews, and Swaminathan Subramaniam who also helped to transcribe the data.

This work was supported in part by the NSF under award number CCF-0448472. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

References

[1] B. Al-Ani and S. E. Sim. So, you think you are a requirements engineer? (extended abstract for poster). In *14th Intl. Requirements Engineering Conference (RE'06)*, 2006.

[2] J. Aranda, S. Easterbrook, and G. Wilson. Requirements in the wild: How small companies do it. In *15th Intl. Requirements Engineering Conference (RE'07)*, pages 39–48, 2007.

[3] Aristotle. Nicomachean Ethics, 350 B.C. (trans. W. D. Ross.

[4] H. Beyer and K. Holtzblatt. *Contextual Design : Defining Customer-Centered Systems*. Morgan Kaufman, 1997.

[5] R. L. Campbell, N. R. Brown, and L. DiBello. The programmer's burden: Developing expertise in programming. In *The Psychology of Expertise: Cognitive Research and Empirical AI*, pages 269–294. Springer-Verlag, 1992.

[6] M. Chechik, B. Devereux, S. Easterbrook, and A. Gurfinkel. Multi-valued symbolic model-checking. *ACM Trans. Softw. Eng. Methodol.*, 12(4):371–408, 2003.

[7] J. M. Corbin and A. C. Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage, 2007.

[8] K. A. Ericsson, R. T. Krampe, and C. Tesch-Römer. The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3):363–406, 1993.

[9] J. Goguen and C. Linde. Techniques for requirements elicitation. *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, pages 152–164, 1993.

[10] L. Hagge and K. Lappe. Sharing requirements engineering experience using patterns. *IEEE Software*, 22(1):24–31, 2005.

[11] D. J. Haraway. *Modest Witness@Second Millenium. FemaleMan Meets OncoMouse: Feminism and Technoscience*. Routledge, 1997.

[12] A. M. Hickey and A. M. Davis. Elicitation technique selection: How do experts do it? In *11th IEEE Joint International Conference on Requirements Engineering (RE'03)*, pages 169–178, 2003.

[13] Intuit, inc. `http://www.intuit.com/`.

[14] J. Lofland and L. H. Lofland. *Analyzing Social Settings: A guide to qualitative observation and analysis*. Wadsworth, 1994.

[15] W. G. Lutters and C. B. Seaman. Revealing actual documentation usage in software maintenance through war stories. *Inf. and Software Technology*, 49(6):576–587, June 2007.

[16] N. Maiden, C. Ncube, and S. Robertson. Can requirements be creative? experiences with an enhanced air space management system. In *28th International Conference on Software Engineering (ICSE '07)*, pages 632–641, May 2007.

[17] B. Nuseibeh and S. Easterbrook. Requirements engineering: A roadmap. In *22nd International Conference on Software Engineering (ICSE '00)*, pages 35–46, June 2000.

[18] C. Potts. Requirements models in context. In *Third IEEE International Symposium on Requirements Engineering (RE'97)*, pages 102–104, 1997.

[19] B. Russell. Knowledge by acquaintance and knowledge by description. *Proceedings of the Aristotelian Society*, 11:108–128, 1910. Reprinted in Russell, Bertrand, Mysticism and Logic, London: Allen and Unwin, 1963, 152-167.

[20] M. Sabetzadeh and S. Easterbrook. View merging in the presence of incompleteness and inconsistency. *Requirements Engineering*, 11(3):174 193, June 2006.

[21] A. G. Sutcliffe and N. A. M. Maiden. Analysing the novice analyst: Cognitive models in software engineering. *Intl. J. of Man-Machine Studies*, 36(5):719–740, May 1992.

[22] S. K. S. Tan. The elements of expertise. *Journal of Physical Education, Recreation and Dance*, 68(2):30–33, 1997.

[23] D. Truex, R. Baskerville, and J. Travis. Amethodical systems development: the deferred meaning of systems development methods. *Accounting Management and Information Technologies*, 10(1):53–79, 2000.

[24] S. Viller and I. Sommerville. Social analysis in the requirements engineering process: From ethnography to method. In *Fourth IEEE International Symposium on Requirements Engineering (RE'99)*, pages 6–13, 1999.

[25] R. Wieringa and C. Ebert. Re'03: Practical requirements engineering solutions. *IEEE Software*, 21(2):16–18, 2004.

[26] R. Wieringa, N. Maiden, N. Mead, and C. Rolland. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering*, 11(1):102–107, 2006.

[27] R. J. Wieringa. Methodologies of requirements engineering research and practice: Position statement. In *First International Workshop on Comparative Evaluation in Requirements Engineering (CERE'03)*, Sept. 2003.