Some Experience Required: *Comparing Professional Requirements Engineering Experience to Students' Perspectives*

Ben Jelen*, Stephen Tsung-Han Sher*, Helena Donaldson*, Gloria Liou*†,
Amanda Stouder*, Steve Chenoweth*, and Mark Hays*
Emails: *jelen@rose-hulman.edu, *sher@rose-hulman.edu, *donaldhm@rose-hulman.edu, *†liou@rose-hulman.edu
*stouder@rose-hulman.edu, *chenowet@rose-hulman.edu, *hays@rose-hulman.edu
*Department of Computer Science and Software Engineering
Rose-Hulman Institute of Technology, Terre Haute, IN 47803
†Department of Psychological Sciences
Purdue University, West Lafayette, IN 47906

***Abstract*—Requirements Engineering (RE) is a challenging topic to teach, especially when not all students see the value in learning it. Students have limited experience with RE before graduation, even after taking a required RE class and doing internships. In contrast, alumni from our program often highlight RE as one of the more important classes they take. To address this gap, we investigated current industry practices and compare that with students' perspectives. We conducted surveys with computer science and software engineering alumni (N=116) and upperclassmen (N=38), asking both groups for their thoughts on RE and examples of where it has shown up in their (budding) careers. We also interviewed survey respondents (N=12) to add more detailed examples of RE. We found that while both alumni and students find the class important, many students have negative impressions of RE and do not have opportunities to gain experience outside of our RE course or capstone. Alumni shared what the current state of RE looks like in practice, such as how some only receive requirements secondhand. Also, their examples highlight how intertwined software engineering areas can be, such as the frequent overlap with software design. We discuss these findings and offer recommendations for teaching requirements engineering to undergraduates, such as giving students RE experiences in the classroom since they do have many other opportunities before graduating.**

I. Introduction

Requirements Engineering (RE) is one of many software engineering skills that are important to teach [1], but RE education remains a challenge. Requirements engineering creates a foundation for the rest of software development, identifying what a project needs to develop to be successful. Often, this involves important steps working with stakeholders to elicit requirements, prioritize them, document them, and validate them [2]. Not all institutions teach dedicated courses on RE [3], despite this importance, which may stem from how challenging it can be to teach. There are many ways of teaching RE, such as role-playing [4], but it is difficult to determine what should be taught and which methods are most effective [5], [6]. One challenge in teaching RE is handling what is unknown and what is unknowable [7] because requirements will not be given as a perfect problem statement [8], [9]. Software engineers' role in requirements gathering may not always be as direct with the increasingly common role of product management [10], posing further challenges to understanding how RE might be evolving.

We were motivated to learn more about RE in practice to improve our undergraduate RE course. Anecdotally, we saw a disconnect between students, who generally did not look favorably on the course, and alumni, who implied it was one of the most important skills they learned at Rose-Hulman. We wanted to understand the factors of this disconnect – How is RE evolving in industry? Is there something missing in how we teach Requirements Engineering? How much, if any, experience is required to appreciate RE?

We conducted a survey of students (N=38) and alumni (N=116) with follow up interviews (N=12) to investigate the perceived discrepancy in opinion of RE. The survey asked participants how they ranked the importance of RE against nine other software engineering skills, their thoughts about RE, examples of how they used RE, and advice for students taking an RE course. The semi-structured interviews asked participants to share specific RE experiences, encouraging them to share details for later analysis. We then analyzed these results to better understand students' and professionals' RE experiences to identify if and why there is a discrepancy between alumni and students' opinions of RE.

We make the following contributions:

1) An understanding of professionals' current experiences with RE in industry, including how it overlaps with similar SE areas,
2) An understanding of students' experience with RE by the time they graduate,
3) A comparison of alumni and students' views of RE.

In the next section, we highlight related work around professional experiences with RE and how it is taught to students. We provide a short background on Rose-Hulman's requirements engineering course. The methods section describes the survey and interview protocols and shares details about participants.

We then share our survey results with accompanying detailed quotes from the interviews. Finally, we discuss these findings, highlighting current RE practices, how alumni and students' views of RE differ, and our recommendations for teaching RE.

II. Related Work

A. Professional Experiences with RE

Several researchers have collaborated together to better understand international Requirements Engineering processes to try to identify underlying theories for how people practice it. The Naming the Pain in RE (NaPiRE) Project [11] is an international collective of researchers who conducted a survey to gather empirical evidence that feeds into an evolving theory of RE practice [12], [13], [14]. For *elicitation*, they found that interviews, facilitated meetings, and prototyping are the most common methods, often with the goal of gathering information from multiple different viewpoints and promoting a shared understanding of requirements. Most RE *documentation* involves free-form textual representations (e.g., written requirements) with fewer using use cases or business process models. RE in industry follows a continuous *change management process* that may start after an initial acceptance period. Most trace requirements to code and/or design documents, and few use systems models to derive formal tests. Many requirements engineers follow a company *standard* for the RE process to help with integrating RE with the development process, such as with testing. Lastly, most companies have project teams continuously *improve their RE process* or task business units with doing so. Wagner et al.'s theory of RE [14], briefly described here, gives some insights across organizations for how practitioners approach RE. We build on this work by providing a broad picture of current practices while comparing both professionals and students' perspectives on requirements engineering.

Several studies have looked at job ads across various countries to see what employers value in Requirements Engineering positions. First, it is rare for RE to be a dedicated role [15], [16], [17] as it is often combined with several other responsibilities, such as quality assurance, deployment, and project or product management. RE jobs also highly value experience [15], [16], [17]. Half of the Canadian job ads reviewed by Wang et al. [17] required 3-5 years of RE experience, and over a fifth required even more. Many employers also looked for experience in non-RE positions, including software engineering, management, and more. Employers expected job seekers to be skilled in specific requirements engineering methods [15], such as requirements management (44%), requirement specification (38%), and quality assurance (37%) [17]. Many job ads also expected applicants to have soft skills, such as communication and writing [15], [16], [17]. These studies shed light on the diverse RE activities employers expect requirements engineering roles to perform, which our study supplements with a broader look at all CS or SE graduates.

Studies also explored the RE process within organizations. Hidellaarachchi et al. [18] followed a development team of 11 members and their RE process to try to understand the role that personality plays. They found that team members scoring higher on facets indicating cooperative, organized and responsible nature, and willingness to change tend to be more comfortable making requirement changes. They also found that assertion levels made some better at managing change requests than others. Franch et al. [19] interviewed participants to explore how different companies use requirement specification and found that many of the same issues that historically have plagued documentation are still there (e.g., ambiguity [20], inconsistency [20], incompleteness). While these studies shed some light on what RE looks like in practice, our work aims to reveal broader RE roles and skills to glean insight into of how CS and SE graduates use RE.

B. RE Education

What content should an RE course focus on? How should assessments be designed to verify RE learning objectives are met? Daun et al. [5] conducted a systematic literature review of requirements engineering education and analyzed approaches used by educators across 152 publications. Their review revealed that elicitation and process modeling were the two most commonly emphasized aspects of RE in the literature. However, the authors also expressed concern over an excessive focus on elicitation and the insufficient representation of software prototyping. Thus, they explored the approaches to student assessment described in the literature. Most instructors reported using course projects, often going to great lengths to replicate industry environments. There was a division between using real external projects and simulated internal projects run by actors to recreate that atmosphere. The authors caution that addressing industry's needs are more important than merely replicating industry's environment [5]. As a response to provide better guidance for future RE course development, this paper examines the requirements activities that industry practitioners are currently engaging in and the ways in which these can be folded into course design.

Researchers have also looked at the current state of RE Education and have strongly advocated for role-playing as a way to give students more experience. Hertz and Spoletini [3] studied SE curricula in the United States and found that RE is infrequently taught. Over half of programs do not have a dedicated RE course, with some only dedicating a few hours in an introductory course. Many are not dedicating time to elicitation and do not use active learning practices, such as role-playing. In contrast, several studies and experience reports have advocated for role-playing to give students experience with RE [4], [21], [22]. They advocated for pulling in stakeholders from industry [4], [21] and showed that role-playing improved students' outcomes on written exams [22]. Our work continues developing an understanding of the current state and highlights the importance of experience in attitudes towards RE courses.

III. Background

Our Requirements Engineering course at Rose-Hulman Institute of Technology was heavily inspired by the work of Callele and Makaroff [23] and we documented iterations in Rupakheti et al. [24]. Callele and Makaroff introduced a variety of classroom activities aimed at rapidly surprising and motivating students to grasp the immediate significance of requirements engineering [23]. Building on this philosophy, previous Rose-Hulman instructors developed homework and term projects that challenged students' preconceptions and emphasized the importance of software requirements engineering [24]. The course highlighted in this paper follows these established traditions.

Our course has evolved significantly over time, but the current version can be summarized as "talk to your clients". Students focus on gaining experience communicating with non-technical clients, eliciting and understanding the Needs (problem domain), Features (proposed solutions), and (Non-)Functional Requirements (detailed requirements someone can write code to) for a client. Students initially elicit requirements through interviews with volunteers role-playing as clients and email follow-up questions to accounts managed by the instructional team. Students then confirm their elicited requirements through prototyping and usability testing, as well as evaluating scope and prioritization based on client needs and expectations. Throughout the course, students are required to communicate with their clients to check their understanding and verify that their final product meets client expectations.

This RE course uses a template project that is significantly outside students' experience, such as a technical field well outside of Computer Science. Having a domain that is foreign to the students ensures that students and the client are not speaking the same language, which helps students all have a similar experience throughout the project and course [24]. Furthermore, care is put into place to ensure students believe that their clients are real industry professionals.

We use an HCI textbook and instructor written PDFs for the course. For HCI aspects, such as interviewing techniques, we rely on "Interaction Design: Beyond Human-Computer Interaction" by Rogers et al [25]. There are not many books written at an undergraduate level for RE, but as we sifted through those available we settled on "Managing Software Requirements" by Leffingwell and Widrig [26]. It used examples that did not expect work experience while covering what we expected. In 2023, we switched to instructor written and edited PDFs as a way to bring the material up to date and reduce students' need to purchase an e-text version of the book.

We still assess our course using specification-based grading on their projects, reading quizzes, homework, and exams. We described how the assessment evolved in section 4 of Rupakheti et al. [24], and the assessments remain the same. The main improvement since then has been that we offer students an ungraded initial submission prior to their graded final submission on each homework and project milestone. These give students feedback to ensure they are on the right track for the final submission. This also simulates the iterative requirements process students will experience when working with clients.

Our RE course is required and we started teaching it in 2003 to juniors. Alumni in our study who graduated in 2004 or later will have taken some version of the course. The current version of the course with the template-based project was started in 2016, thus alumni who graduated 2018 or later will have taken the contemporary version of the course.

IV. Methods

This study applied a mixed methods approach, using both surveys and semi-structured interviews to explore the perspectives of alumni and students. The surveys are the primary data source, and the interviews supplement participants' views with in-depth examples. This research received ethical approval from our institutional review board.

A. Survey

We surveyed 116 alumni and 38 students for (1) their opinions on where RE ranked compared to other software engineering (SE) skills, (2) their thoughts on RE, (3) examples of using RE, and (4) advice for anyone taking a class on RE. We received 165 RE examples from alumni and 18 from students. We also asked respondents if they would be willing to participate in an interview. The surveys for each group are available in the Supplemental Materials[1].

We used a small deception to establish unbiased opinions when ranking SE skills by advertising that the survey wanted to understand how respondents felt about SE skills broadly, before revealing that most questions were focused on RE. The 10 SE skills participants ranked were derived from the Software Engineering Education Knowledge (SEEK) from the SE Curriculum Guide [27]. Participants were given a definition of each skill from SEEK as part of the survey (see Supplemental Materials[1]).

We recruited alumni through an alumni email listserv, and we recruited students either through an email blast or by requesting participation in person from students in RE classes. Everyone filled out the survey online. Respondents were excluded when they did not fill out the majority of the survey. Participants were compensated by being entered into a raffle for multiple $25 gift cards.

The participants were all alumni or students from Rose-Hulman Institute of Technology. Alumni (N=116) largely held computer science and/or software engineering degrees (Table I), held software developer positions (Table II), and on average graduated in 2005 (median=2008). Students (N=38) were seniors (N=18) or juniors enrolled in a required requirements engineering course (N=20). All students were computer science and/or software engineering majors. Unless otherwise specified, "students" hereafter refers to the combined group of seniors and juniors.

[1]https://osf.io/4kaw2/

TABLE I
ALUMNI DEGREES. NOTE THAT SOME HELD MULTIPLE DEGREES AND CATEGORIES LIKE "CS ONLY" ARE INCLUDED IN "CS".

Major	Count	Percent (%)
CS only	68	58.6
SE only	16	13.8
EE only	1	0.9
CS	95	81.9
SE	30	25.9
Math	10	8.6
CPE	5	4.3
DS	1	0.9
CS & SE	11	9.5
More than one major	29	25.0
Didn't say	2	1.7
TOTAL	**116**	

TABLE II
CATEGORIZED JOB TITLES OF ALUMNI

Categorized Job Titles	Count	Percent (%)
Senior Developer	43	37.1
Developer	31	26.7
Technical Management	21	18.1
Senior Leadership	9	7.8
Product Management	3	2.6
Other	8	6.9
Other Management	1	0.9
TOTAL	**116**	

B. Interviews

We interviewed participants (N=12) who had indicated in the survey that they were willing to be interviewed (Table III). We recruited participants based on the quality of their open-ended responses, focusing on ones who would likely provide a more diverse set of examples (e.g., one senior who appreciated the RE course and one who did not). We interviewed nine alumni and three senior students. We had hoped to interview more students, especially Juniors taking our RE course, but none responded.

TABLE III
INTERVIEW PARTICIPANT DETAILS

PID	Job Category	Major(s)	Grad. Year
A018	Technical Management	EE	1987
A020	Senior Leadership	CS	1990
A038	Developer	CS	2000
A051	Product Management	CS	2005
A071	Senior Developer	CS & SE	2011
A078	Technical Management	SE	2013
A088	Technical Management	SE	2016
A106	Developer	CS & CompE	2021
A107	Developer	CS	2022
S30	Senior Student	CS	2024
S36	Senior Student	SE	2024
S38	Senior Student	CS	2024

The semi-structured interviews first asked participants to share specific instances of when they used RE, how important RE is to their role, and what advice they would give to students taking a class on RE. Probing questions further elicited details about the participants' reactions and internal dialogue. Two initial alumni interviews focused on the survey responses, but we then adjusted our interview protocol to emphasize more on collecting examples of RE practice. All interviews were anonymized and transcribed. In our results section, we use these interviews to add richer detail to the survey responses.

Interviews averaged 31 minutes and were conducted over video conferencing or phone. All interview participants were compensated with gift cards ($20 for alumni and $10 for students).

C. Analysis

1) Survey Analysis: We calculated the mean rankings of the aforementioned 10 SE skills to compare alumni and students' responses about the relative importance of requirements engineering skills for succeeding in their career. For alumni responses, we conducted additional analysis. We examined descriptive statistics for frequency of using requirements engineering, change in frequency, and thoughts around the importance of having a required requirements engineering course. We used correlation analysis to examine whether there was a linear relationship between frequency of using requirements engineering and thoughts around the importance of having a required requirements engineering course. Additionally, we used one-way analysis of variance (ANOVA) to examine differences in frequency of use, frequency change, and importance of having a required course based on job title and cohort (i.e., years since graduation).

Qualitative survey data – examples of RE, advice to students, job title categorization – was categorized using inductive, bottom-up analysis until we had enough consensus to code using deductive, top-down approaches (e.g., [28]). Each qualitative question was first coded by one author, and any concerns or borderline codes were discussed with another author.

We categorized the requirements engineering activities in the RE examples based on definitions from the Software Engineering Body of Knowledge (SWEBOK) v3 [2][2]. We focused on picking the top-level requirements engineering activities listed in SWEBOK (Figure 1.1 on page 1-2), but some were subcategories. We chose subcategories when they were commonly discussed in examples (e.g., Requirements Tracing is a subcategory of Practical Considerations). Some codes were formed based on bottom-up analysis, including: whether they mentioned needing to clarify requirements, who they implied they worked with to define requirements, whether software design was discussed, and whether product management was discussed.

2) Interview Analysis: The interviews are supplemental findings with more detailed examples than we received in surveys. Because of participants' diverse experiences, we did not reach saturation and thus did not perform rigorous analysis (e.g., grounded theory or thematic analysis). Instead, we transcribed the interviews and reviewed each for participant stories that exemplified our survey findings.

[2]SWEBOK v4 was not available yet at the time of analysis

D. Limitations

Participants in our study are only from Rose-Hulman Institute of Technology. We are a Science, Technology, Engineering, and Math (STEM) focused institution in the United States, so our alumni and students' experiences are not a broad representation of software engineering experiences. We felt this trade-off was worth recruiting a large, age-diverse alumni participant pool. Further, we started teaching an RE course in 2003, so about half of the alumni had no formal training. That may have influenced their understanding of what RE is and the advice they would give students.

Many of our results hinge on participants' response to the following question about examples of RE in practice, "Share an example of when you used software requirements engineering in your current job. What, if any, methods or tools did you use? Write 'N/A' if you don't use it in your job.". The survey did not explicitly ask specific questions that arose from our bottom-up analysis, such as whether they needed to spend time clarifying requirements or which specific methods they used. These results, therefore, should be read as the first thought to come to mind for participants, but not necessarily an exact count of how frequently any of those activities happen in practice.

We conducted a cohort statistical analysis to look for differences across generations of alumni, but this test works best with longitudinal studies. The survey only took a snapshot of many different alumni at one point in time rather than following them over the long term.

Lastly, our findings would be enriched by understanding the size of the company alumni worked for, but we did not ask for that data in our survey. For example, we suspect having someone else handle talking to end users (e.g., product management roles) may be more prevalent in larger companies.

V. Results

We share our results highlighting alumni's professional experiences with RE, students' thoughts about it, and several comparisons between them.

A. Both Say RE Education Is Important

Both alumni and students acknowledge that RE is an important part of CS and SE undergraduate education. They rated the importance an average of four out of five (Alumni: 4.03; Students 3.95).

RE was in the top half of SE Skill rankings for both groups, but alumni ranked it higher than students. Alumni found it more important with an average rank of 4.53, putting it 3rd out of 10 behind Software Design (3.83), and Computing Essentials (4.48). Students ranked it 5th (5.24) behind software design (4.18), computing essentials (4.39), software quality (4.78), and software verification and validation (5.18). The only statistically significant difference between the two groups was that students ranked software verification and validation (5.19) as more important than alumni (6.33; $P<.05$; $P=.011$).

Alumni who do RE find RE undergraduate education valuable. Those who more frequently report using RE in their jobs were also more likely to say that taking an RE class is important for undergraduates ($R=.23$; $P<.05$)[3].

B. Both Say They Use RE or Expect to

Both alumni and students report they used or expected to use RE at least a few times per year. Most alumni (103/116; 89%) reported using RE at least "a few times a year", and most students (16/18; 89%) expected to use RE at least a few times a year in their future careers. There was not a statistically significant difference between the two groups.

When we split the alumni into cohorts, we found differences in whether they use RE more or less than when they started. Younger alumni were more likely to report using RE more since they started their jobs, highlighting how they are gaining experience with RE. That plateaued with the 2004-2013 cohort before increasing again for older cohorts (Figure 2). Participants were asked to rate how RE skill use changed since they began their jobs, with 1 meaning it decreased significantly, and 5 meaning it increased significantly. Viewing this data in cohorts shows a statistically significant trend – 2019-2024 was 3.58 and 2014-2018 was 3.50 before dipping down to close to "stayed the same" (3.03) for 2004-2013, and rising slightly again for 1977-2003 (3.26).

C. Elicitation is the Main RE Activity

When asked for examples of using RE, elicitation was the most common RE activity for both alumni (100/165; 60.6%) and students (17/18; 94.4%), although not everyone reported doing so (Figure 1). Other activities in the SWEBOK, such as requirements tracing, were not mentioned nearly as frequently by alumni and students.

The majority of participants did not mention a specific elicitation method, instead stating generally that they spoke to clients, such as A107 saying *"I had to consult with [clients] on their needs and features for the system."* The majority of alumni that talked about eliciting requirements did not specify a specific elicitation method (56/100; 56.0%), but some gave more specifics in their examples. Process modeling (20/100; 20.0%), interviews (17/100; 17.0%), and prototyping (13/100; 13.0%) were the most common, and a quarter (27/100; 27.0%) mentioned multiple methods. Students also did not frequently specify an elicitation method (11/17; 64.7%), but they did mention interviews (4/17; 23.5%) as the most common method.

D. Alumni See Overlap between RE and Software Design

There was heavy overlap between RE and Software Design in practice, with some alumni seeming to struggle to distinguish the two. Nearly half of alumni examples (74/165; 44.9%) included references to software design strategies, such as incremental design approaches with quick feedback cycles. For example:

> *I helped the team to develop "Tech Briefs" which were limited to one-page of technical design – using*

[3]The question implied taking *any* RE course is important, not necessarily ours since not all alumni took our RE course.

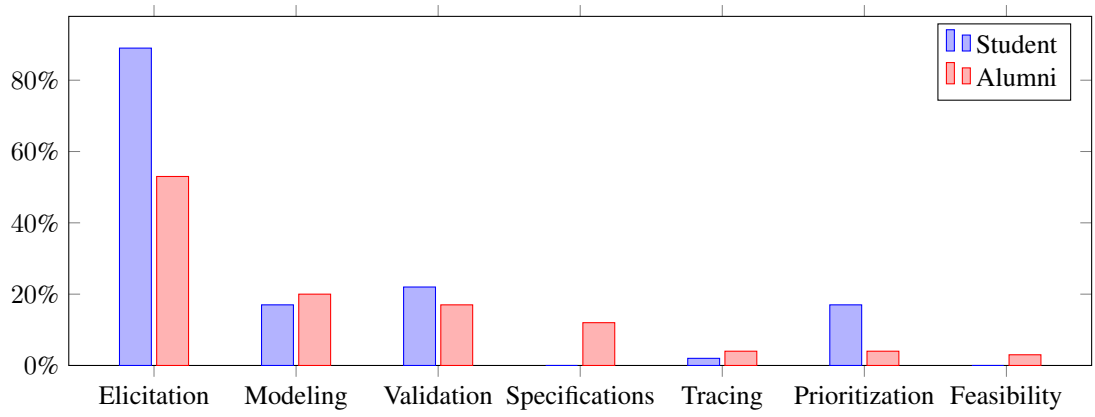

Fig. 1. Most Common RE Activities for Alumni and Students When Asked How They Use RE. Elicitation is the most common for both.

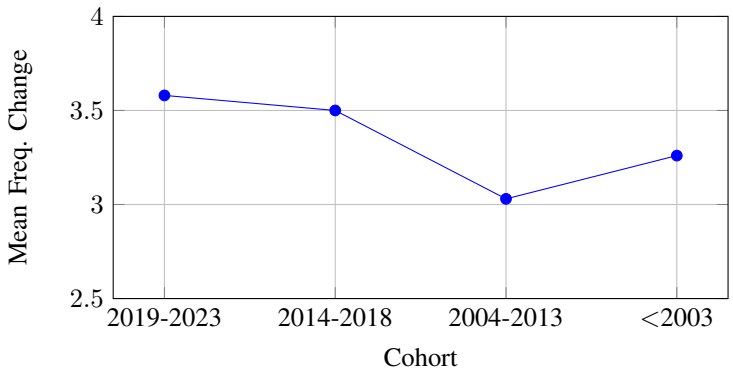

Fig. 2. Rating of Whether Using RE More or Less Since Starting Job. One was decreased significantly and five was increased significantly. Younger cohorts said they were increasingly using RE. 2004-2013 plateaued, before older cohorts used RE more.

> *diagrams or bullet item lists to quickly capture a design and get feedback from others either within the team or from customers/users of the tech. We wrote very few "Full Technical Design Documents" because while they were helpful for the writer to collect thoughts – nobody else used them.* (A034, senior developer, survey)

A034's emphasis on their use of "Tech Briefs" demonstrates how technical development work is often meshed with software design practices. Even though we provided a definition of RE (drawn from SWEBOK v3 [2]), some alumni still appeared to misinterpret RE *as* software design. Some (21/165; 12.7%) *only* mentioned software design considerations in their RE examples. For example, A031 (product manager, survey) shared, *"I assisted a developer in working with the Project Manager to implement a collection of reports [to use] shared libraries that implemented expected development standards and performance, while segmenting out portions that needed an extra level of security."*. There was not any statistically significant difference between cohorts of alumni and the way in which they interpreted RE and Software Design.

TABLE IV
COUNTS OF WHO ALUMNI WERE TALKING TO IN THEIR RE EXAMPLES WHEN SOMEONE WAS AT LEAST IMPLIED. *PERCENTAGES ARE PERCENTAGES OF EXAMPLES IN THIS TABLE, SO THEY ARE OF THOSE WHERE SOMEONE WAS AT LEAST IMPLIED. NOTE THAT SOME EXAMPLES MENTIONED MULTIPLE CATEGORIES.

Categories	**Count**	**Percent (%)***
External Clients	35	36.5
Internals Who Talk to External Clients	22	22.9
Internal Clients	19	19.8
Participant Answer Questions	6	6.3
Implied, but Unspecified	14	14.6
Total examples with a person implied	**96**	

E. Alumni Are Not Always Talking Directly to Customers

In practice, alumni often talk to internal or external clients (54/96; 56.3%), but a fair number of participants report hearing requirements second hand (22/96; 22.9%) (Table IV). These alumni who were not involved with elicitation tended to attribute a "Product Manager" or similar "higher-up" to that circumstance:

> *For more than a decade, I've basically never talked to end users when I've operated as a Software Engineer. Requirements are gathered by "Product Managers" who work with visual designers and*

> *users to figure out the initial requirements. Engineering needs to make it as fast, cheap, and maintainable as possible, negotiating with product management on those requirements when something is particular[ly] expensive in time, cost, or maintainability. For example: what can be cut? what can be done differently? and what is a nice-to-have vs a real requirement? I think engineering has become "refiners" of requirements more than "definers" of requirements.* (A054, senior developer, survey)

Ultimately, 12.1% (20/165) of alumni examples mentioned product management being involved, and 2.6% of alumni (3/116) worked as product managers in their current job.

Not directly talking to customers required more vigilance from alumni, often resulting in time spent clarifying requirements. Similar to A054, some alumni mentioned that RE in practice often meant needing to clarifying requirements (23/165; 13.9%), and each of those alumni (23/23) were working with managers or product managers to define requirements or they did not specify. In our interviews, this division of labor was not necessarily seen as an issue but a necessary organization. A038 (Developer, interview) shared how they felt it worked well to have a manager conduct the requirements gathering because they were often excellent communicators and translators of customer requirements.

F. Many Students Come to RE Courses with Concerns

Confirming our anecdotes that students come in disliking our RE course, juniors starting to take it often come in hearing negative or neutral opinions about the course from other sources, which were usually focused on procedures rather than RE content. When we classified what they had heard about the course, almost half had heard negative comments (8/20; 40%), such as, *"A lot of students hate the course. They don't like the content and may not see it useful"* (S17, junior, survey) or *"It [is] one of the harder classes the CS student will have to take."* (S05, junior, survey). Almost half had heard neutral comments (8/20; 40%), such as *"Between [RE] and [Software Design], any given student will find one very difficult (in the top 5 hardest classes) and the other average difficulty"* (S15, junior, survey) and *"This class contains a lot of work. Not usually very difficult work, but a lot of work"* (S19, junior survey). One student did share positive comments (1/20; 5%), such as, *"It was easy"* (S13, junior, survey). Others (3/20; 15%) reported having not heard anything about the course. We had hoped to interview some students about how these comments had influenced their views of the course, but none responded to our requests for an interview.

G. Other than an RE Course, Students Only Gain RE Experience in Senior Project

Students were only getting experience with RE through our dedicated course and senior project (i.e., Capstone), not from senior research project or any internships. Most Senior Project (i.e., Capstone) students (11/13; 84.6%) indicated that they used RE techniques as they worked on their year-long group design projects:

> *Yes, we interviewed our client to acquire a basic list of needs and features, expanded those into requirements and then use cases, then created test cases for each with a traceability matrix.* (S33, senior, survey)

Only one of the Senior Thesis students (1/5; 20%) reported using RE, although they primarily used RE skills and not an RE process directly:

> *In my thesis, I had to design a workshop mostly focused on understanding the needs of domestic abuse survivors and eventually making some recommendations about a robot. I think that RE helped me design a good set of questions for each session and more confidence with working with non-technical people.* (S26, senior, survey)

The majority of graduating seniors (12/18; 66.7%) reported that they *did not* use RE in their internships. Some students elaborated:

> *As an intern, I was never responsible for communicating with the client. Instead, a full-time employee would translate the client's needs into a description which I would then implement. This was more similar to a basic [CS or SE] class where requirements were given and I would write the code to satisfy them.* (S32, senior, survey)

Three students with internships (3/18; 16.7%) reported eliciting requirements from stakeholders but not conducting any other RE activities. Instead, they often engaged in informal conversation about requirements with *"...manager[s] and people who would use the software tool* (S30, senior, survey). The remaining three students (3/18; 16.7%) reported conducting various RE activities, such as managing *"pain points"*, *"feature prioritization"*, *"release schedules"*, and other client requests (S36, senior, survey).

H. Students Give Procedural Advice, while Alumni Give Advice on RE Practice

The majority of students emphasized *procedural* advice for how to succeed in an RE course rather than *practical* advice on doing better RE. Their advice centered on two primary time management strategies: starting homework early and allocating sufficient time to communicate with the client (Figure 3). Anecdotally, this mirrors reflection questions after our RE course, when we ask students how they could improve in the course. A common piece of advice from students regarded the importance of starting homework early:

> *Prioritize this class. This should be the first homework you are working on, because it's not like other classes where you can pull an all-nighter and finish it. Instead, it actually takes a couple hours each day, from the very beginning.* (S32, senior, survey)

Other advice included typical examples of recommending that future students *"communicate often"* and *"pay attention to client details"* (S25, senior, survey).

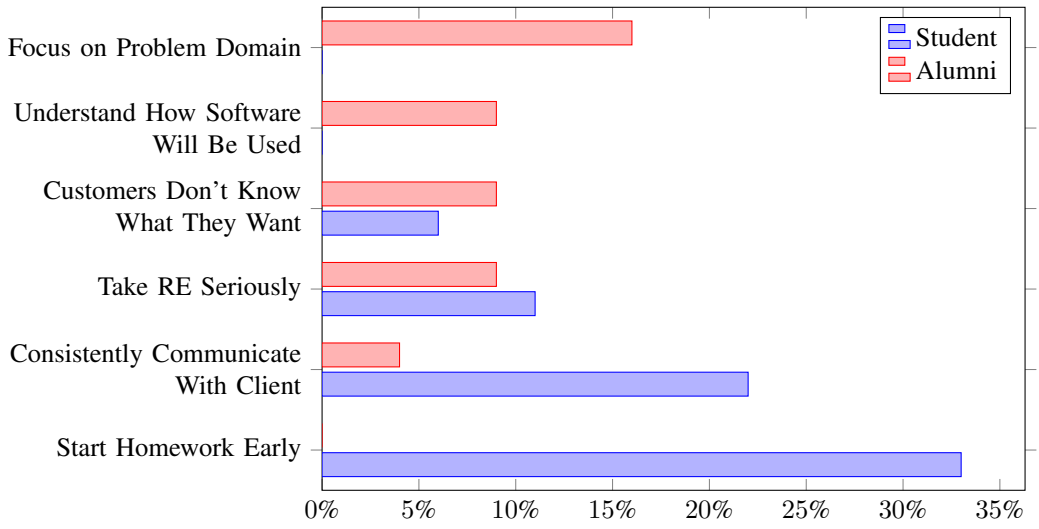

Fig. 3. Most Common RE Course Advice. Students advised to start homework early 33% of the time. Alumni gave more varied answers, advising to focus on the problem domain 16% of the time.

Alumni provided more detailed advice regarding the *practice* of RE. The most frequent recommendation (19/116, 16.4%) was for students to concentrate on thoroughly understanding the problem domain:

> *Realize that people don't pay you to code, they pay you to solve their problems with code. Learn the skills now, so that you can solve their problems easily in the future so that you can quickly return to whatever pet project you have waiting where you are your own boss.* (A061, senior developer, survey)

Alumni also consistently provided detailed responses on the significance of RE (10/116; 8.6%), such as:

> *The formal process of gathering requirements is extremely crucial to most small businesses. Often this skill and knowledge is severely lacking in smaller companies without due processes. Good requirements literally make or break your product and potentially the company's success, so it is extremely important to understand how and why requirements and needs are determined as well as a documented process to follow through.* (A105, developer, survey)

VI. Discussion

Our results help us explain the anecdotes that motivated this study: Why is it that alumni say Requirements Engineering is an important course, yet students tend to have a negative view of the course? We will discuss what RE looks like for our alumni – a messy and indirect process at times. We found that alumni and students both find RE important, but without as much experience as alumni, students often come into RE courses with a negative outlook. We end by offering some implications for teaching RE.

A. RE Is Messy in Practice

Most developers would agree that software engineering, especially at scale, is a convoluted process. Requirements engineering is one way developers provide some structure to the process, focusing on defining the problem and thinking through a plan before developing. However, RE in industry can be messy [29]. Through our research, we found some commonalities to RE in practice, but we also highlight what contributes to the messiness of RE.

There were commonalities in the RE activities participants highlighted, and they were most often talking directly to clients. Elicitation was the most common RE activity in examples – when people think of requirements engineering, they often think of eliciting requirements. Modeling, validation, and specifications were reported in at least 10% of examples. Frequently, the CS and SE alumni from our institution spoke directly to internal or external clients. These findings mirror what others have found about RE in practice – elicitation is the most common RE activity [12], [14], other RE activities occur but not as frequently [14], and typically RE involves talking to clients to obtain high-quality requirements [12]. At the very least, CS and SE students should expect to elicit requirements from end users.

Through alumni's examples, we suspect that RE is messier in practice, in part from the blurred line between RE and software design. A majority of participants highlighted part of the software design process in their examples of RE. This should not be a surprise – both factor heavily into the early stages of the development process. In some organizations, the person gathering requirements might also be the person designing the software and writing the code. For others, roles may be divided up. The agile framework even expects that developers will frequently transition between RE and software design as a system comes to fruition [30], [31]. What was surprising in our results was how many RE examples *only* mentioned software design (12.7%). Some of that may be explained by participants not reading the RE definition – the survey had a small deception obfuscating the focus on RE after

all – but that is still a high proportion of examples. Perhaps participants view them as so intertwined that they cannot be separated, or they do not see a meaningful difference between the two in the same way SE experts do in SWEBOK.

Another factor contributing to the messiness of RE is the division of labor in practice. A quarter of alumni said they work with someone else who speaks to customers and develops requirements. As developers become further removed from the source of the requirements, we would expect a quality or coordination cost to the software being developed. We saw the coordination cost from several participants who mentioned how they have to spend time clarifying requirements (13.9%). Often, a dedicated product manager is responsible for determining the requirements and/or maintaining relationships with users. Product managers can prevent failures in translating those needs into a cohesive vision for a product [32]. Product management is not a new concept [33] – a growing number of companies rely on such roles to translate needs into requirements and find success in doing so. However, this could potentially lead to requirement misinterpretations or at least a coordination cost on members of the development team.

B. Students' Lack of RE Experience Likely Explains Their Dislike for RE Course

Both alumni and students agree that Requirements Engineering is not the most important software engineering domain, but it is an important one to consider for the success of CS and SE graduates. Participants ranking RE in the top half of important SE skills, but not necessarily the most important skill, further bolsters Garousi et al.'s [1] meta analysis of SE skills required by industry. Garousi et al. found that in more recent papers (2013-2018) RE was considered important but near the middle of the pack. Both groups in our study also used or expected to use RE in their careers at rates that are not significantly different from each other. Our work further supports the value of RE.

Our study focuses on the experience disparity between alumni and students. This disparity likely helps explain why anecdotal appreciation for RE might grow with experience. We saw how alumni increasingly used RE in their jobs compared to when they started, especially in the first 10 years. Those who use RE more frequently find RE courses more important to take. They also gave advice that focused more on RE content, rather than procedural details of how to do better in an RE course.

Students are missing the appreciation for RE that comes with practical experience. Researchers have shown how engaging students with practical examples in the classroom can improve their interest and motivation in learning RE [34], [35], [36], but real world experience can help cement their learning. At our institution, students take our required project-based RE course and a majority of students then go on to take Senior Project where they gain experience with RE. However, students reported that they were not exposed to RE during internships. We had expected that more would gain some experience there, but it is clear that is not true. Instead, most do not see RE outside the classroom until they enter the workforce. Our work implies that regardless of the extent to which we incorporate RE in curricula, some students inevitably will only develop an appreciation for RE after they gain experience in industry, as role-played clients and RE scenarios cannot fully replace actual industry practice.

C. Recommendations for Teaching RE

First, we recommend that **educators continue to focus efforts on teaching requirements elicitation [5], but they may want to place more emphasis on introducing other RE activities**. Again, elicitation was the most common RE activity mentioned, so programs similar to ours that emphasize elicitation already align with what happens in practice. Indeed, Daun et al.'s [5] systematic review of RE Education research also emphasizes the importance of elicitation over many other RE activities. Alumni's frequent use of interviews as their primary elicitation method lends credence to starting there, as does Wagner et al.'s broad survey of RE practitioners [14]. An added benefit that we found at our institution is that interview skills are useful in some of the other methods mentioned by participants, such as facilitated meetings.

We validated the anecdotes in our data – students start our RE course having heard generally negative comments about it, yet alumni conduct and appreciate RE in their careers. One change to our thinking from prior to this study is that students appreciate and expect to use RE in practice, too – they do not significantly differ from alumni. We take this finding as a reminder that undergraduates' anecdotal opinions of a course are not always based on the content, but can also reflect procedural aspects, such as how difficult the course is [37]. As SE educators, we should not necessarily expect students to appreciate everything we identify as important in the curriculum, such as ethics, process, and requirements [38]. However, this study implies that students' lack of RE experience may be playing a larger role in the negative anecdotes we hear from students. Therefore, our second recommendation is that **educators teach RE knowing that an appreciation for RE might not happen until they gain practical experience**. For example, teaching elicitation techniques and introducing other RE activities could be a valuable model for instilling practical RE skills that they will appreciate later. Following this recommendation can be difficult for an instructor, especially when student evaluations are one of the few sources of feedback on how we are doing as educators.

Finally, our third and fourth recommendations are that **educators continue giving students RE experiences in the classroom because they are not getting those experiences outside the classroom**, and **educators should encourage students to ask to do more RE in internships**. There are some RE skills, such as interviewing [3], that cannot be taught purely through theory and discussion. Our work adds to the chorus of support for supporting RE experiences [5], [14], but we caution against assuming external experiences, like internships, are likely to expose students to the role of RE in their future careers. We hope that encouraging students to

advocate to participate in RE will help give them more RE experiences, but we acknowledge that it would be challenging for employers given that experience is seen as key to good requirements engineering [15], [16], [17]. External clients could absolutely still be a valuable source of experience as many suggest [4], [21], [22], but those experiences should be structured to promote giving students the opportunities to experience RE. At Rose-Hulman, our RE course is setup to provide students with those experiences by supporting a templated external client[4]. This mirrors the philosophy we adopted from other programs, such as University of Texas at Dallas [23]. Further, most seniors at Rose-Hulman gain more experience in RE during their senior project. They work in a group of four for an external client, but faculty (many of whom also have taught our RE course) also advise students to emphasize good practices in RE.

VII. Conclusion

We sought to examine differing perspectives on RE between professionals and undergraduate computing students. Our surveys, completed by 116 alumni and 38 upperclassmen from our institution, gathered their opinions on various SE skills (emphasizing questions on RE) while encouraging participants to describe how RE tools and activities were used in their past and current positions. Using follow-up interviews conducted with 12 alumni and student survey participants, we further explored participants' opinions on the importance of RE in the SE curriculum, RE's impact on their current or future career, and specific experiences that molded their viewpoint.

We found that alumni and students both recognized the importance of RE courses within a CS or SE curriculum, and both stated elicitation as a common RE activity. In practice, alumni described experiences with RE that highlighted how convoluted it can be, such as working with secondhand requirements and how hard it was for them to cleanly separate RE from other SE areas, such as software design. We confirmed that students expressed apprehension about taking our institution's RE course, and we found that they were not getting experience in RE during their internships, only during formal classes.

We discussed how these findings highlight how RE is messy in practice, and students' lack of RE experience in professional settings may help explain why students dislike our RE course. We recommend that SE educators continue to teach RE skills and activities, knowing that students simply might not appreciate it until they gain practical experience outside the classroom. SE educators should also continue constructing opportunities for CS and SE students to gain experience in the classroom since they are not getting chances in internships. We should continue to emphasize elicitation to mirror practice, but we may want to introduce other RE activities as well. Lastly, educators should encourage students to advocate for gaining experience with RE during internships.

[4]Please see the Background section for more details

VIII. Data Availability

Our data is unavailable. We did not ask participants if we could share their responses during the informed consent process. With so many responses, it is nearly impossible to obtain consent from all participants after the fact. Additionally, our ethics board was concerned that even if we anonymized survey results and interview transcripts, we may still jeopardize participants' privacy.

Acknowledgments

We want to first thank our participants for helping us learn more about their perspectives on requirements engineering. We also want to thank Rose-Hulman's Alumni Relations Office for helping with recruiting alumni, and Rose-Hulman's Office of Institutional Research, Planning and Assessment for their help with conducting the survey.

References

[1] V. Garousi, G. Giray, E. Tüzün, C. Catal, and M. Felderer, "Aligning software engineering education with industrial needs: A meta-analysis," *Journal of Systems and Software*, vol. 156, pp. 65–83, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121219301347

[2] P. Bourque and R. E. Fairley, "Guide to the software engineering body of knowledge," *IEEE Computer Society Press*, 2014. [Online]. Available: https://www.computer.org/education/bodies-of-knowledge/software-engineering/v3

[3] K. Hertz and P. Spoletini, "Are requirements engineering courses covering what industry needs? a preliminary analysis of the united states situation," in *2018 IEEE 8th International Workshop on Requirements Engineering Education and Training (REET)*, 2018, pp. 20–23.

[4] G. Gabrysiak, H. Giese, and A. Seibel, "Why should i help you to teach requirements engineering?" in *2011 6th International Workshop on Requirements Engineering Education and Training*, 2011, pp. 9–13.

[5] M. Daun, A. M. Grubb, V. Stenkova, and B. Tenbergen, "A systematic literature review of requirements engineering education," *Requirements Engineering*, vol. 28, no. 2, pp. 145–175, Jun. 2023. [Online]. Available: https://link.springer.com/10.1007/s00766-022-00381-9

[6] O. Dieste and N. Juristo, "Systematic review and aggregation of empirical studies on elicitation techniques," *IEEE Transactions on Software Engineering*, vol. 37, no. 2, pp. 283–304, 2011.

[7] R. J. Barnes, D. C. Gause, and E. C. Way, "Teaching the unknown and the unknowable in requirements engineering education," in *2008 Requirements Engineering Education and Training*, 2008, pp. 30–37.

[8] R. L. Quintanilla Portugal, P. Engiel, J. Pivatelli, and J. C. S. do Prado Leite, "Facing the challenges of teaching requirements engineering," in *Proceedings of the 38th International Conference on Software Engineering Companion*, ser. ICSE '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 461–470. [Online]. Available: https://doi.org/10.1145/2889160.2889200

[9] K. Sikkel and M. Daneva, "Getting the client into the loop in information system modelling courses," in *2011 6th International Workshop on Requirements Engineering Education and Training*, 2011, pp. 1–4.

[10] C. Ebert and S. Brinkkemper, "Software product management – an industry evaluation," *Journal of Systems and Software*, vol. 95, pp. 10–18, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121214000156

[11] D. M. Fernández, S. Wagner, M. Felderer, and M. Kalinowski, "The naming the pain in requirements engineering (napire) project." [Online]. Available: http://www.napire.org/

[12] D. M. Fernández, S. Wagner, M. Kalinowski, A. Schekelmann, A. Tuzcu, T. Conte, R. Spinola, and R. Prikladnicki, "Naming the pain in requirements engineering: Comparing practices in brazil and germany," *IEEE Software*, vol. 32, no. 5, pp. 16–23, 2015.

[13] D. Méndez Fernández and S. Wagner, "Naming the pain in requirements engineering: A design for a global family of surveys and first results from germany," *Information and Software Technology*, vol. 57, pp. 616–643, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584914001256

[14] S. Wagner, D. M. Fernández, M. Felderer, A. Vetrò, M. Kalinowski, R. Wieringa, D. Pfahl, T. Conte, M.-T. Christiansson, D. Greer, C. Lassenius, T. Männistö, M. Nayebi, M. Oivo, B. Penzenstadler, R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. Spínola, A. Tuzcu, J. L. D. L. Vara, and D. Winkler, "Status quo in requirements engineering: A theory and a global family of surveys," *ACM Trans. Softw. Eng. Methodol.*, vol. 28, no. 2, feb 2019. [Online]. Available: https://doi.org/10.1145/3306607

[15] M. Daneva, C. Wang, and P. Hoener, "What the job market wants from requirements engineers? an empirical analysis of online job ads from the netherlands," in *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '17. IEEE Press, 2017, p. 448–453. [Online]. Available: https://doi.org/10.1109/ESEM.2017.60

[16] A. Herrmann, "Requirements engineering in practice: there is no requirements engineer position," in *Proceedings of the 19th International Conference on Requirements Engineering: Foundation for Software Quality*, ser. REFSQ'13. Berlin, Heidelberg: Springer-Verlag, 2013, p. 347–361.

[17] C. Wang, P. Cui, M. Daneva, and M. Kassab, "Understanding what industry wants from requirements engineers: an exploration of re jobs in canada," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3239235.3268916

[18] D. Hidellaarachchi, J. Grundy, R. Hoda, and I. Mueller, "What's personality got to do with it? a case study on the impact of personality on requirements engineering-related activities." in *Proceedings of the 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering*, ser. CHASE '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 182–194. [Online]. Available: https://doi.org/10.1145/3641822.3641874

[19] X. Franch, C. Palomares, C. Quer, P. Chatzipetrou, and T. Gorschek, "The state-of-practice in requirements specification: an extended interview study at 12 companies," *Requir. Eng.*, vol. 28, no. 3, p. 377–409, apr 2023. [Online]. Available: https://doi.org/10.1007/s00766-023-00399-7

[20] S. Hotomski, E. B. Charrada, and M. Glinz, "An exploratory study on handling requirements and acceptance test documentation in industry," in *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 2016, pp. 116–125.

[21] B. Penzenstadler, M. Mahaux, and P. Heymans, "University meets industry: Calling in real stakeholders," in *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*, 2013, pp. 1–10.

[22] R. B. Svensson and B. Regnell, "Is role playing in requirements engineering education increasing learning outcome?" *Requir. Eng.*, vol. 22, no. 4, p. 475–489, nov 2017. [Online]. Available: https://doi.org/10.1007/s00766-016-0248-4

[23] D. Callele and D. Makaroff, "Teaching requirements engineering to an unsuspecting audience," in *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 433–437. [Online]. Available: https://doi.org/10.1145/1121341.1121475

[24] C. R. Rupakheti, M. Hays, S. Mohan, S. Chenoweth, and A. Stouder, "On a pursuit for perfecting an undergraduate requirements engineering course," *Journal of Systems and Software*, vol. 144, pp. 366–381, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121218301390

[25] Y. Rogers, H. Sharp, and J. Preece, *Interaction Design: Beyond Human-Computer Interaction*, 6th ed. USA: Wylie, 2024.

[26] D. Leffingwell and D. Widrig, *Managing Software Requirements: A Use Case Approach*, 2nd ed. USA: Pearson Education, 2003.

[27] A. f. C. M. Joint Task Force on Computing Curricula IEEE Computer Society, "Software engineering 2014: Curriculum guidelines for undergraduate degree programs in software engineering," *Computing Curricula Series*, 2015. [Online]. Available: https://www.acm.org/binaries/content/assets/education/se2014.pdf

[28] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[29] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Information Systems Journal*, vol. 20, no. 5, pp. 449–480, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2575.2007.00259.x

[30] A. Alliance, "What is agile?" [Online]. Available: https://www.agilealliance.org/agile101/

[31] R. C. Martin, *Agile Software Development: Principles, Patterns, and Practices*. USA: Prentice Hall PTR, 2003.

[32] M. A. Cusumano and D. B. Yoffie, *Competing on Internet time: Lessons from Netscape and its battle with Microsoft*. The Free Press, 1998.

[33] C. Ebert and S. Brinkkemper, "Software product management – an industry evaluation," *Journal of Systems and Software*, vol. 95, pp. 10–18, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121214000156

[34] M. Daun, A. Salmon, B. Tenbergen, T. Weyer, and K. Pohl, "Industrial case studies in graduate requirements engineering courses: The impact on student motivation," in *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T)*, 2014, pp. 3–12.

[35] L. Mich *et al.*, "Teaching requirements analysis: A student project framework to bridge the gap between business analysis and software engineering." in *2014 8th International Workshop on Requirements Engineering Education and Training (REET)*, 2014, pp. 20–25. [Online]. Available: https://ceur-ws.org/Vol-1217/paper3.pdf

[36] B. Tenbergen and M. Daun, "Industry projects in requirements engineering education: application in a university course in the us and comparison with germany," in *Hawaii International Conference on System Sciences 2019 (HICSS 2019)*, 2019.

[37] H. W. Marsh, "The influence of student, course, and instructor characteristics in evaluations of university teaching," *American Educational Research Journal*, vol. 17, no. 2, pp. 219–237, 1980. [Online]. Available: http://www.jstor.org/stable/1162484

[38] T. C. Lethbridge, J. Diaz-Herrera, R. J. J. LeBlanc, and J. B. Thompson, "Improving software practice through education: Challenges and future trends," in *Future of Software Engineering (FOSE '07)*, 2007, pp. 12–28.