The Journal of Systems and Software 219 (2025) 112200

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Attributes of a great requirements engineer[☆]

Larissa Barbosa [a], Sávio Freire [b], Rita S.P. Maciel [a], Manoel Mendonça [a], Marcos Kalinowski [c], Zadia Codabux [d], Rodrigo Spínola [e,*]

[a] Federal University of Bahia, Salvador, Bahia, Brazil
[b] Federal Institute of Ceará, Morada Nova, Ceará, Brazil
[c] Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil
[d] University of Saskatchewan, Saskatoon, Canada
[e] Virginia Commonwealth University, Richmond, VA, United States

ARTICLE INFO

Keywords:
Requirements Engineering
Attributes of a great requirements engineer
Skill
Survey

ABSTRACT

Context and motivation: Several studies have investigated attributes of great software practitioners. However, the investigation of such attributes is still missing in Requirements Engineering (RE). The current knowledge on attributes of great software practitioners might not be easily translated to the context of RE because its activities are, usually, less technical and more human-centered than other software engineering activities.
Question/problem: This work aims to investigate which are the attributes of great requirements engineers, the relationship between them, and strategies that can be employed to obtain these attributes. We follow a method composed of a survey with 18 practitioners and follow up interviews with 11 of them.
Principal ideas/results: *Investigative ability in talking to stakeholders*, *judicious*, and *understand the business* are the most commonly mentioned attributes amongst the set of 22 attributes identified, which were grouped into four categories. We also found 38 strategies to improve RE skills. Examples are *training*, *talking to all stakeholders*, and *acquiring domain knowledge*.
Contribution: The attributes, their categories, and relationships are organized into a map. The relations between attributes and strategies are represented in a Sankey diagram. Software practitioners can use our findings to improve their understanding about the role and responsibilities of requirements engineers.

1. Introduction

Several studies have investigated attributes of great software practitioners (Buckingham, 2005; Hemon et al., 2020; Rivera-Ibarra et al., 2010; Li et al., 2015; Kalliamvakou et al., 2019; Dias et al., 2021; Gren and Ralph, 2022). For instance, Li et al. (2015) identified 53 attributes of great software engineers, such as adaptable, knowledge about people and the organization, and creativity. The authors also divided the attributes into categories, revealing that the attributes can be either internal, i.e., related to engineer's personality and ability to make effective decisions, or external, i.e., associated with the impact that great engineers have on people and software products. Another work investigated attributes of great software maintainers of open-source projects, revealing that technical excellence and communication are the most recurring attributes (Dias et al., 2021). Knowing these results is necessary to help growing better software engineers and maintainers in this ever-demanding software industry.

Nevertheless, we believe current information about the attributes of great software practitioners might not be easily translated to the context of requirements engineering (RE). RE is a human intensive area. While several software engineering activities such as analysis, design, coding, and testing tend to make use of automated processes, activities related to RE are more related to social interactions among stakeholders involved in software development processes (Pohl and Rupp, 2015). Requirements engineers are "a translator that understands the domain as well as its particular language well enough and also possesses enough IT know-how to be aware of the problems the developers face and to be able to communicate with them on the same level" (Pohl and Rupp, 2015).

In the book *Requirements Engineering Fundamentals*, Pohl and Rupp (2015) mentioned that analytic thinking, empathy, communication skills, conflict resolution skills, moderation skills, self-confidence, and persuasiveness are expected attributes of a requirements engineer.

[☆] Editor: Prof Neil Ernst.
[*] Corresponding author.
E-mail addresses: larissa.leoncio@ufba.br (L. Barbosa), savio.freire@ifce.edu.br (S. Freire), rita.suzana@ufba.br (R.S.P. Maciel), manoel.mendonca@ufba.br (M. Mendonça), kalinowski@inf.puc-rio.br (M. Kalinowski), zadiacodabux@ieee.org (Z. Codabux), spinolaro@vcu.edu (R. Spínola).

https://doi.org/10.1016/j.jss.2024.112200
Received 8 September 2023; Received in revised form 17 August 2024; Accepted 27 August 2024
Available online 5 September 2024

Another text book of the area, *Handbook for the Advanced Level Requirements Elicitation* (Häuß er et al., 2022), describes the following attributes: communication, courtesy, flexibility, integrity, interpersonal skills, positive attitude, professionalism, responsibility, teamwork, and work ethic. These attributes, somehow, support our guess that attributes of requirements engineers have their own particularities. However, in both text books, there is a lack of empirical evidence supporting the described list of attributes. Indeed, to the best of our knowledge, there is not an investigation on attributes of a great requirements engineer. The lack of empirical evidence, in our understanding, hinders researchers from reasoning about them, employers from identifying them, and young engineers from becoming them.

Software RE is a determinant of productivity and product quality and it is inherently complex due to influences from customer environment, making the requirements engineer a central role in software development processes (Damian and Chisan, 2006; Fernández et al., 2017). There is a need to understand which attributes of requirements engineers are considered relevant in software development organizations in present day and how to pursue them. This is especially relevant, for instance, from the perspective of graduates and other novice software engineers seeking and recruited to requirements engineering jobs. Better understanding of the current desirable attributes can also be utilized in revising and improving RE education.

In this paper, we investigate the attributes of great requirements engineers and the relationship between them. To guide our research, we designed the following research questions (RQs):

- **RQ1**: What are the attributes of a great requirements engineer?
- **RQ2**: How these attributes are related to each other?
- **RQ3**: What strategies can be used to pursue these attributes?
- **RQ4:** How practitioners perceive the importance of the identified strategies to pursue the attributes of great requirements engineers?

We use a research method composed of a survey and follow up interviews. Through the survey, we ask practitioners up to five attributes that are most important for requirements engineers. After, we interview a subset of those who responded the survey to gather more information about the mentioned attributes. Lastly, we asked the interviewed participants about the level of importance of the identified strategies to pursue the attributes of a great requirements engineer. The level of importance is related to the extent to which a strategy can support a requirements engineer to improve her/his skills to get closer to being a great requirements engineer.

We surveyed 18 Brazilian software practitioners and 11 of them accepted our invitation for an interview. Based on their answers, this paper's contributions include (i) a list of 22 attributes of great requirements engineers, (ii) a map which relates those attributes, and (iii) a set of 38 strategies that can be employed to pursue them. Software practitioners can use the map to improve their understanding about the role and responsibilities of requirements engineers. The identified strategies work as tools to improve their skills.

Besides this introduction, this paper is organized in six other sections. Section 2 presents related work. Then, Section 3 presents the data collection and analysis procedures. Section 4 presents the results of the survey and follow up interview. Section 5 discusses the main findings. Section 6 presents the threats to the study validity. Lastly, Section 7 presents our final remarks and the next steps of this work.

2. Related work

Much of our knowledge about desirable attributes of software practitioners comes from studying specific roles such as software engineer, project manager, and software manager. Li et al. (2015) investigated attributes of great software engineers by interviewing experienced engineers. They reported 53 attributes of great software engineers, which were classified into two categories, internal and external. Examples of identified attributes are improving, knowledgeable about people and the organization, and creative.

In another study, Kalliamvakou et al. (2019) performed a two-phase study to investigate the attributes of a great manager of software engineers. By interviewing software practitioners from Microsoft, the authors collected the perception of predefined attributes identified in technical literature or used by Microsoft. Afterwards, a survey was grounded from the interview results to identify the attributes' importance. The authors identified 15 attributes organized into a conceptual framework. Furthermore, the attributes were divided into two dimensions (the levels of interaction and the engineering manager's functions) and categories (team, individual, manager cultivating engineering wisdom, motivating the engineers, and mediating communication). Maintains a positive working environment, grows talent, and enables autonomy were the most important attributes from the point of view of the survey participants.

More recently, Dias et al. (2021) investigated the attributes of great open-source software maintainers. By conducting an interview study with open-source software maintainers, the authors identified 22 attributes. After, by surveying contributors of open-source software, the authors collected qualitative information about the importance of these attributes. Communication, quality assurance, and community building are considered the most important attributes. The authors also organized the attributes in a conceptual framework, showing the relationship between them.

Lastly, Gren and Ralph (2022) surveyed agile professionals to identify which makes effective leadership in agile software teams. The authors identified ten categories (e.g., step in if needed, build a strong team identity, and adapt to customer culture) divided into three themes (dynamic team leadership, social identity, and organizational culture). These categories and themes were organized in a model of agile leadership.

We recognize the valuable results reported by related work. Indeed, we were inspired by them to define our research method based on survey and follow up interviews (see Section 3). However, it is still missing the mapping of attributes of great requirements engineers. This is precisely the topic we approach in this paper.

We compare related work with our results in Section 5.2 to investigate if our initial guess ("We believe current information about the attributes of great software practitioners might not be easily translated to the context of requirements engineering") holds.

3. Research method

We followed a method based on the work of Kalliamvakou et al. (2019) and Dias et al. (2021). For answering the research questions, we surveyed software practitioners to identify the attributes of great requirements engineers (RQ1), interviewed a subset of them to gather more details about each attribute (RQ2 and RQ3), and asked the interviewed participants a follow up question to identify the level of importance of the strategies to become a great requirements engineer (RQ4). The following subsections present the data collection and analysis procedures for each step of the research method.

3.1. Survey on attributes of great requirements engineers

3.1.1. Design and data collection

We invited requirements engineers and project managers from our Brazilian industrial partners. We considered these two roles to capture multiple perspectives: those responsible to performing RE activities and those who manage them.

The survey comprises a set of characterization questions and an open-ended question for participants indicate up to five attributes that are most important for performing RE activities. We asked only up to five attributes because individuals usually pay more attention

Table 1
Survey questions.

ID	Survey question (SQ) description	Type
SQ1	What is your name?	Open-ended
SQ2	Please, inform us your e-mail address, so we can contact you in the next step of the study.	Open-ended
SQ3	How would you describe your gender?	Closed
SQ4	How many years of experience do you have with software requirements?	Open-ended
SQ5	How many companies have you worked for?	Open-ended
SQ6	What is the size of your company (software and other areas)?	Closed
SQ7	What project role are you assigned in this company?	Closed
SQ8	How do you rate your experience in this role?	Closed
SQ9	Based on your experience, what attributes of a software requirements engineer would you rate as the attributes that matter most to being a great requirements engineer? Please enter up to five attributes ordered by their level of importance, with the highest importance listed first.	Open-ended

Table 2
Categories adapted from Dias et al. (2021).

Name	Description
Management	This category refers to attributes that help managing a project, in the sense of understanding the vision of the project, establishing and communicating project goals, proposing timelines, managing the quality of documentation, etc.
Personality	This category refers to how practitioners express what they think, feel, and behave in their interaction with other project members, referring to personality aspects.
Social	This category groups attributes related to how practitioners deal with other project members.
Technical	This category is about technical skills that are used to perform requirements engineering activities.

to the top few choices, mitigating additional noise in lower rankings (Ben-Akiva et al., 1992). Focusing on just five attributes would (i) keep participants from running their response, (ii) require them to provide focused responses, discarding less relevant attributes, and (iii) reduce the cognitive load of remembering specific past experiences. Table 1 presents the survey questions and their types (closed or open-ended).

3.1.2. Data analysis

We performed different data analysis procedures, as the survey is composed of closed and open-ended questions. For closed questions, we calculated the number of participants choosing an option. After, we summarize the participants' characterization. For open-ended questions, we analyzed the list of attributes reported by the participants looking for similarities, i.e., two or more attributes have the same meaning. For example, the attributes understand the business and knowledge of the business problem were unified under *understand the business* attribute. First and second authors analyzed individually the list to standardize the attributes. Divergences were resolved by the last author. This process was necessary because we did not provide a pre-determined list of attributes for the participants.

We also perceived that the attributes are related to each other, allowing us to group them into categories. The categories are based on those defined by Dias et al. (2021) (see Table 2). For example, we used *social* category to group the attributes *good listener*, *investigative ability in talking to stakeholders*, *clarity*, and *knowledge about user behavior*. The grouping process was performed using constant comparison (Strauss and Corbin, 1998) and was performed by the first author and reviewed by the last author.

Table 3
Interview script.

Section	Interview question (IQ) description
Opening	We present the informed consent term. **IQ1.** In your words, how would you define a great software requirements engineer? **IQ2.** Have you ever worked with a great software requirements engineer? **IQ3.** Why did you consider him/her a great requirements engineer?
Attributes for a great requirements engineer	In the survey, you listed five desirable attributes for a great requirements engineer. Now, let us address three of them: <<*attributes a, b, and c*>>. **IQ4.** How would you define *attribute a*? **IQ5.** Why is *attribute a* important for great software requirements engineers? **IQ6.** What are the strategies to obtain *attribute a*? **IQ7.** How can a requirements engineer use *attribute a*? **IQ8–IQ11.** Same questions for *attribute b*. **IQ12–IQ15.** Same questions for *attribute c*.

3.2. Follow up interview

3.2.1. Design and data collection

We defined the interview script (Table 3) comprised of two sections. In the **opening** section, the participant provided his/her definition for great software requirements engineer and described characteristics of a great one that he/she have worked with. In **attributes of a great software requirements engineer** section, the participant defines, explains the importance, gives strategies to achieve, and exemplifies the use of three attributes. Although each participant has informed up to five attributes in the survey, we choose the three highest-ranked attributes from her/his list.

We carried out the interviews remotely. Each of them lasted around 30 min and were recorded with the participant's permission.

3.2.2. Data analysis

We transcribed the interviews and organized the answers by questions. Then, we coded the transcriptions to identify the main idea presented in each answer (Seaman, 1999; Strauss and Corbin, 1998). For example, the participant *P1* reported the following answer to IQ1: "the issue of communication is important, because it has to be a person who knows to listen and intervene at the right times (soft skills), and from a technical point of view, it is more important to know the domain." In this example, we found the following codes: *communication*, *listening skills*, *intervention at appropriate time*, *interpersonal skills*, and *domain knowledge*. In IQ6, the respondents indicated strategies to obtain a specific attribute. For example: "knowing how to listen, knowing how to write, passing on the information that the client needs to whoever is developing (P7)." From this response, we identified the following strategies: *knowing how to listen, improving writing, and sharing the customer's need with the development team.* The coding process was performed by the first and second authors. Each of them, individually, coded half of the transcripts and then revised each other. Afterwards, they met with the last author to resolve the divergences.

The codes extracted from the definition of attributes reported by participants in IQ4 are **concepts**. We applied conceptual mapping over the concepts to identify relationships between the attributes (see Fig. 1). For example, we identified that the concept *conduct meetings with customers and users* is present in the definition of the attributes *translate user needs into software requirements* and *good knowledge of requirements engineering practices*. Thus, we assumed that these attributes are related to each other. Both of them depends somehow on conduct meetings with customers and users. The first and second authors performed this process, and met with the last author to review the results.

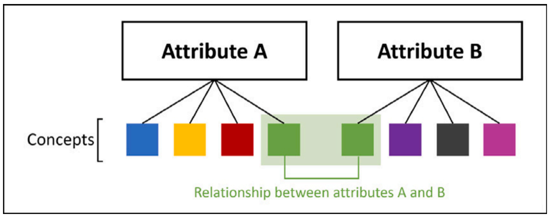

Fig. 1. Analysis schema.

3.3. Follow up question on the importance of the identified strategies to pursue the attributes of great requirements engineers

3.3.1. Design and data collection

We performed this step after obtaining the results from the follow up interviews. To determine the level of importance of each strategy we identified during the interviews (IQ6), we asked a follow up question (see Table 4) to the interviewed participants. For each strategy, we presented its definition and asked the participant to indicate how important it is for a requirements engineer to improve her/his skills to get closer to being a great requirements engineer, considering a 5-point Likert scale (strongly disagree, disagree, neither agree nor disagree, agree, and strongly agree).

3.3.2. Data analysis

We used descriptive statistics to analyze the answers by counting the share of participants choosing each option available to rank the strategies.

4. Results

In total, we invited 18 Brazilian software practitioners and they completely answered the survey, but only 11 of them were available for interviews and for answering the follow up question. We used answers from the survey (18 answers) to answer RQ1, while the interviews redand answers for the follow up question were used to answer RQ2-3 and RQ4, respectively.

4.1. Demographics

Table 5 presents the participants' characterization indicating their years of experience, the number of organizations the participant has worked, current company size (small — up to 50 employees; medium-sized — with 51 to 1000 employees; large — more than 1000 employees), and role. All participants are from different organizations.

We can notice that most of the participants are requirements engineers and most have from eleven to 20 years of level of experience in their roles. Most of the participants work in medium-sized companies and most have worked in at least six different companies in their professional life.

Although our sample consists of software practitioners from the Brazilian software industry, it includes a diverse group of professionals with varying levels of experience in project management or requirements engineering roles across companies of different sizes. Details of this diversity can be observed in Table 5.

4.2. RQ1: What are the attributes of a great requirements engineer?

We identified 22 attributes of a great requirements engineer. Table 6 presents the five most cited attributes, along with the total number of participants that cite an attribute (#CA) and the percentage of the total of mentions (%A). The complete list of attributes is available in Barbosa et al. (2023). *Investigative ability in talking to stakeholders* was the most mentioned attribute, followed by *judicious*, *understand the business*, *good ability to identify missing requirements*, and *good knowledge of requirements engineering practices*. These attributes represent 55% of all mentioned attributes frequency.

Table 4
Follow up questionnaire.

ID	Follow up question (FQ) Description	Type
–	We identified 38 strategies that could be used to enhance the attributes of a great requirements engineer. Below, we present these strategies. For each of them, please indicate how important it is for a requirements engineer to become a great requirements engineer.	–
FQ1	**Participating in training** Improving the skills of the requirements engineer through courses, reading specialized material and getting to know new methodologies.	Closed
FQ2-FQ38	...	Closed

From the interviews, we identified the definition of the attributes. *Investigative ability in talking to stakeholders* attribute means that a software requirements engineer must be prepared not to limit themselves to what the interested party is saying, but to try to provoke them for answers about the system requirements, as described by the participant P3 “so you have to prepare, study, create a script, so I think that this investigative capacity comes from there...” *Judicious* attribute refers to the ability to organize the software documentation and pay attention to details, for instance, P6: “As a very detail-oriented person, he not only investigate but detail the information obtained so that it becomes clear in documentation and easy to communicate.”

Understand the business attribute is related to understand the environment where the system will be used, as described by P1: “It would be the understanding of the business context, where that system will be inserted. Nothing from a technical point of view, from the business itself, from the environment...” *Good ability to identify missing requirements* attribute is associated with the ability to propose solutions considering the user context and validate them with customers, as illustrated by P5: “the analyst needs to be able to understand when something is not clear, when things are not going well. And after, having the ability to elicit what is missing and the ability to validate that with the customer.” Lastly, *good knowledge of requirements engineering practices* attribute refers to have knowledge on the practices used to elicit, analyze, and specify software requirements. For instance, participant P1 reported “understanding the technique behind these elements that we use... to get in touch with users...”

By constantly comparing the codes (Strauss and Corbin, 1998), we grouped them into the following categories:

- **Personality**: It has seven attributes to express how requirements engineers think, feel, and behave in their interaction with stakeholders. These attributes are: *agility*, *analytical reasoning*, *antifragility*, *associative reasoning*, *commitment*, *curiosity*, and *judicious*.
- **Management**: It groups the following attributes to support managing a project or the requirements: *good ability to identify missing requirements*, *good systematic view*, *understand the business*, *update requirements (when necessary)*, and *set priorities with customer*.
- **Technical**: It is composed of seven attributes associated with technical skills that are used in software requirements activities. These attributes are: *ability to validate implemented requirements*, *good writing (knowing how to write clearly and without ambiguities)*, *experience*, *good knowledge of general software development practices*, *good knowledge of requirements engineering practices*, *prior knowledge, even if basic, in some programming language*, and *translate user needs into software requirements*.
- **Social**: It encompasses the following attributes related to how requirements engineers deal with stakeholders: *clarity*, *good listener*, *investigative ability in talking to stakeholders*, and *knowledge about user behavior*.

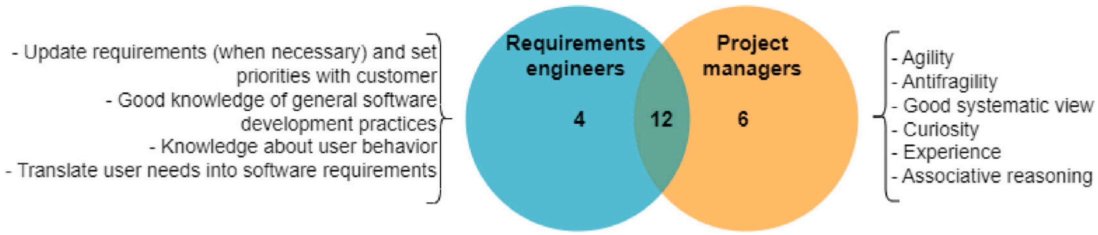
- Update requirements (when necessary) and set priorities with customer
- Good knowledge of general software development practices
- Knowledge about user behavior
- Translate user needs into software requirements
Requirements engineers
4
12
Project managers
6
- Agility
- Antifragility
- Good systematic view
- Curiosity
- Experience
- Associative reasoning

Fig. 2. Venn diagram comparing attributes from requirements engineers' and project managers' perspective.

Table 5
Participant's demographics.

Participant ID	Years of experience	#Companies worked	Company size	Role	Follow up Interviews (Y/N)
P1	15	4	Large	Project manager	Yes
P2	2	3	Small	Requirements engineer	Yes
P3	10	6	Large	Requirements engineer	Yes
P4	25	5	Large	Project manager	Yes
P5	3	4	Small	Requirements engineer	Yes
P6	2	3	Medium-sized	Requirements engineer	Yes
P7	16	7	Large	Requirements engineer	Yes
P8	5	10	Large	Project manager	Yes
P9	5	8	Medium-sized	Requirements engineer	Yes
P10	30	11	Large	Project manager	Yes
P11	10	11	Medium-sized	Project manager	Yes
P12	15	5	Large	Project manager	No
P13	1.5	1	Medium-sized	Requirements engineer	No
P14	15	3	Large	Project manager	No
P15	12	5	Large	Requirements engineer	No
P16	1	1	Medium-sized	Requirements engineer	No
P17	20	10	Large	Project manager	No
P18	1	4	Medium-sized	Requirements engineer	No

Table 6
The five most cited attributes.

Attribute	#CA	%A
Investigative ability in talking to stakeholders	15	18%
Judicious	12	14%
Understand the business	7	8%
Good ability to identify missing requirements	7	8%
Good knowledge of requirements engineering practices	5	6%

Caption:
#CA: Number of mentions of an attribute.
%A: Percentage of #CA in relation to the total all mentioned attributes (83).

Table 7
Categories of attributes.

Category	#A	#CA	%A
Personality	7	25	30%
Social	4	24	29%
Management	5	18	22%
Technical	6	16	19%

Caption:
#A: Number of unique mentioned attributes.
#CA: Number of mentions of an attribute.
%A: Percentage of #CA in relation to the total all mentioned attributes (83).

Table 7 presents the categories of attributes, reporting the category's name, the number of unique attributes mentioned (#A) and the total number (i.e., count) of participants that cite attributes in each category (#CA). Lastly, the column %A corresponds to the percentage of #CA in relation to all mentioned attributes. The categories personality and social are the most mentioned, highlighting the central skills of a great requirements engineer.

4.2.1. Difference in perception of the attributes between requirements engineers and project managers

As our population is composed of requirements engineers and project managers, we investigated whether they have a difference in their perception of the attributes of a great software requirements engineer. Fig. 2 shows a Venn diagram representing the result of the comparison. We identified six attributes only reported by project managers. These attributes are related to soft skills that increase the requirements engineer's ability to perform their activities. We also recognized four attributes reported only by requirements engineers. These attributes are associated with hard skills needed to perform software requirements activities. Finally, twelve attributes were reported by both.

4.3. RQ2: How these attributes are related to each other?

In total, we identified 13 relationships between attributes. Table 8 shows these relationships along with their attributes, concepts, and categories. A concept is the definition part shared by two or more attributes. For example, the concept *be investigative to define and associate the problems* is part of the definition of the following attributes *analytical reasoning, associative reasoning*, and *judicious*, as we can see in "P8: It's the ability you have to see a problem. And being able to find requirements to define that problem," "P8: It's you being able to associate a problem you're seeing today, and problems you've seen in the past," and "P6: As a very detail-oriented person, he not only investigate but also detail the information obtained so that it becomes clear," respectively. In Table 8, we only show the attributes that have at least one relationship with other attribute.

We organized these concepts and the relationships in a map (Fig. 6) which is further discussed in Section 5.1.

4.4. RQ3: What strategies can be used to obtain these attributes?

From the interviews, we identified 38 strategies to obtain the attributes of a great requirements engineer. Table 9 presents the five most cited strategies, along with the total number (i.e., count) of participants

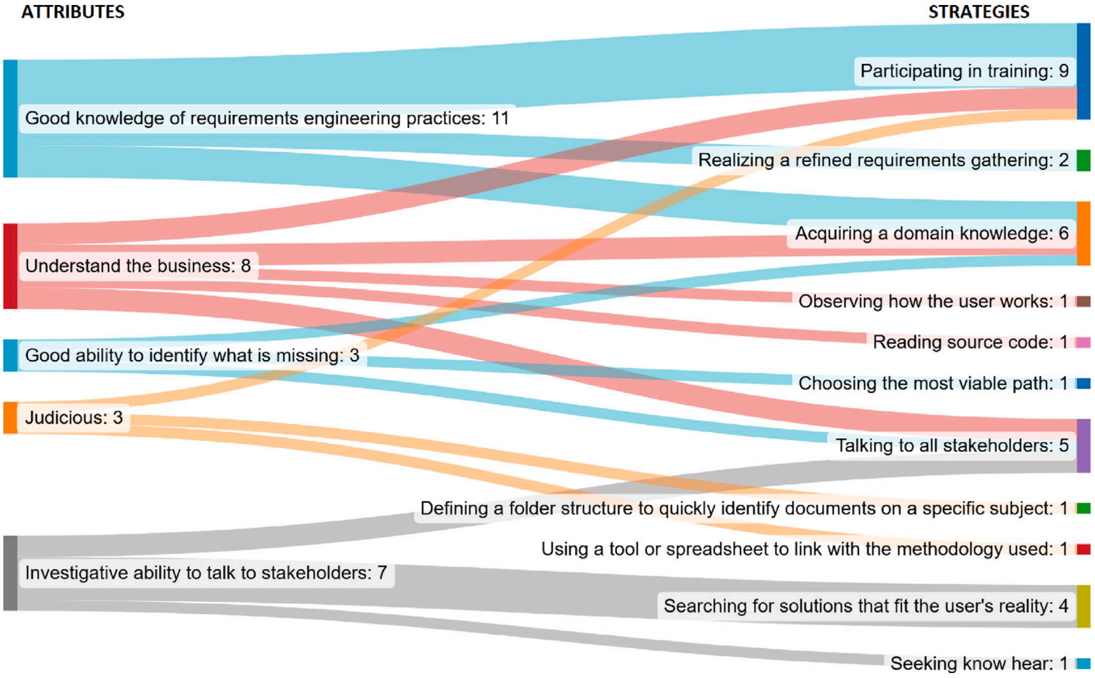
ATTRIBUTES
STRATEGIES
Good knowledge of requirements engineering practices: 11
Understand the business: 8
Good ability to identify what is missing: 3
Judicious: 3
Investigative ability to talk to stakeholders: 7
Participating in training: 9
Realizing a refined requirements gathering: 2
Acquiring a domain knowledge: 6
Observing how the user works: 1
Reading source code: 1
Choosing the most viable path: 1
Talking to all stakeholders: 5
Defining a folder structure to quickly identify documents on a specific subject: 1
Using a tool or spreadsheet to link with the methodology used: 1
Searching for solutions that fit the user's reality: 4
Seeking know hear: 1

Fig. 3. Sankey diagram for the most mentioned attributes and the top three strategies to obtain them.

Table 8
Relationships between the attributes.

Category	Attribute relationships	Concept
Management	Good ability to identify missing requirements ↔ Understand the business	Ability to understand the information flow
	Good ability to identify missing requirements ↔ Understand the business	Ability to understand user regulation
	Good ability to identify missing requirements ↔ Understand the business	Ability to understand business concept
Personality	Associative reasoning ↔ Agility	Turn around problems quickly
	Agility ↔ Judicious	Keep information up to date by understanding the business
	Analytical reasoning ↔ Associative reasoning ↔ Judicious	Be investigative to define and associate the problems
Social	Clarity ↔ Investigative ability in talking to stakeholders	Refine the collected information
	Clarity ↔ Knowledge about user behavior	Refine the collected information
	Good listener ↔ Investigative ability in talking to stakeholders	Do not interfere with the customer’s speech
	Good listener ↔ Investigative ability in talking to stakeholders	Do not think you know everything
	Good listener ↔ Investigative ability in talking to stakeholders	Listen to extract the most information
Technical	Translate user needs into software requirements ↔ Good knowledge of requirements engineering practices	Conduct meetings with customers and users
	Translate user needs into software requirements ↔ Good knowledge of requirements engineering practices	Apply requirements elicitation techniques to understand the users’ needs
	Good knowledge of general software development practices ↔ Experience	Know the areas of RE

that cite a strategy (#CS) and the percentage of the total of mentions (%S). The complete list of strategies is available in Barbosa et al. (2023). The most mentioned strategy was blue*Participating in training*, followed by *acquiring domain knowledge*, *talking to all interested parties*, *searching for solutions that fit the user’s reality*, and *acquiring knowledge about tests*.

Participating in training means improving the skills of the requirements engineer through courses, reading specialized material and getting to know new methodologies. For instance, “P1: Theoretical knowledge, which teaches how to apply the techniques, what is relevant, what kind of attitude the analyst has to have, how he has to prepare, the points of attention he has to have.” *Acquiring domain knowledge* refers to knowledge about a field for which the system is under developed, as described in “P1: studying that area in which the system is inserted” and “P5: knowing your (system) domain.”

Talking to all stakeholders strategy is related to keep in touch with customers, users, and software teams to understand and explain the requirements, such as “P7: Talking, demonstrating their ideas, talking both with the customer and the software developers, listening, exchanging information.” *Searching for solutions that fit the user’s reality* strategy is associated with defining a solution considering users’ needs and restrictions, as illustrated in “P9: it is necessary to seek an innovative, low-cost solution that is developed in the shortest possible time.” Lastly, *Acquiring knowledge about tests* means the ability to define test cases to verify the system in relation to its requirements, for

Table 9
The five most cited strategies.

Strategy	#CS	%S
Participating in training	17	19%
Acquiring domain knowledge	9	10%
Talking to all stakeholders	7	8%
Searching for solutions that fit the user’s reality	4	4%
Acquiring knowledge on tests	4	4%

Caption:
#CS: Number of mentions of a strategy.
%S: Percentage of #CS in relation to the total all mentioned strategies (91).

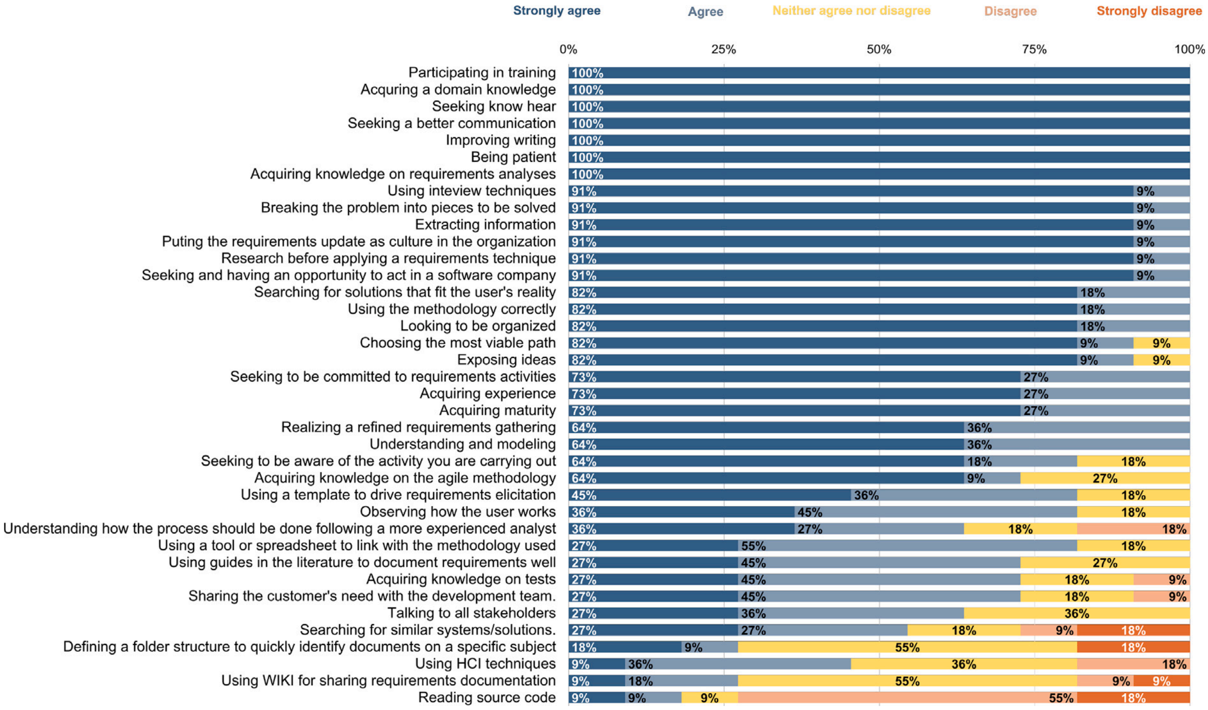

Fig. 4. Interviewees' point of view on the importance of the identified strategies.

example, the participant P11 reported "...create test objectives, create test sections..., know test techniques."

We also investigated the relationships between the strategies and attributes, organizing them in a Sankey diagram (Lupton and Allwood, 2017). This diagram is composed of bars, representing the sources and destination of information, and links, showing the magnitude of the flow between those bars. Fig. 3 shows the diagram for the five most mentioned attributes and the three most mentioned strategies to obtain them. A complete version is available in Barbosa et al. (2023). The flow starts with attributes which is related to strategies. The numeric values appear with each element (attribute and strategy) show the total number of times that a relationship occurs. The thickness of each link varies according to the value of the relationship. For example, the *participating in training* strategy can be used to obtain the attributes *understand the business*, *good knowledge of requirements engineering practices*, and *judicious*.

4.5. RQ4: How practitioners perceive the importance of the identified strategies to pursue the attributes of great requirements engineers?

Fig. 4 presents the results from the follow up question answered by all interviewees. In general, the strategies were well evaluated. *Participating in training*, *acquiring domain knowledge*, *seeking know hear*, *seeking a better communication*, *improving writing*, *being patient*, and *acquiring knowledge on requirements analyses* were considered the main strategies to improve the ability of requirements engineers. On the other side, *searching for similar systems/solutions*, *defining a folder structure to quickly identify documents on a specific subject*, *using HCI techniques*, *using WIKI for sharing requirements documentation*, and *reading source code* are only mentioned by some of the participants.

5. Discussion

This section presents the map that organizes the attributes and their concepts and relationships. Also, it compares our findings with related work and discusses the implications of our results for researchers, managers, novice requirements engineers, and educators.

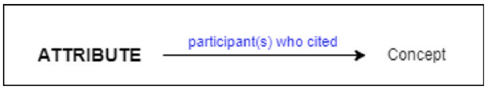

Fig. 5. Elements of the map.

5.1. Map of great requirement engineers

We organized the set of concepts related to each attribute and their relationships in a map. Inspired by the framework structure proposed by Dias et al. (2021), attributes are represented in bold upper-case text, whereas the concepts are represented as smaller, lowercase text. An attribute has a set of concepts that describe it. An arrow relates an attribute and its concepts, and has a blue text to indicate the participant who mentioned the concepts. Fig. 5 exemplifies how our map is structured.

Fig. 6 presents the four maps. We define a map for each attribute category. We can notice that all attributes from management (A) category is related to at least one other attribute. From personality (B) and technical (D) categories, we found relationships for almost all attributes, except *commitment* and *good knowledge of general software development practices* attributes. In the social (C) category, the attributes *clarity*, *investigate ability in talking to stakeholders*, and *know about user behavior* are related to each other. Lastly, the maps do not present *antifragility*, *good systematic view*, *prior knowledge, even if basic, in some programming language*, and *curiosity* attributes, because they were not covered in the interviews.

We also found core concepts (highlighted in green), i.e., a concept that relates two or more attributes. For example, *well-defined and updated documentation* (from Management category) is a concept that relates two attributes *update requirements (when necessary) and set priorities with customer*, and *understand the business*.

Software professionals can use the map to begin or improve their initiatives to address these attributes. As a reference, software professionals can understand how attributes are defined and seek strategies or

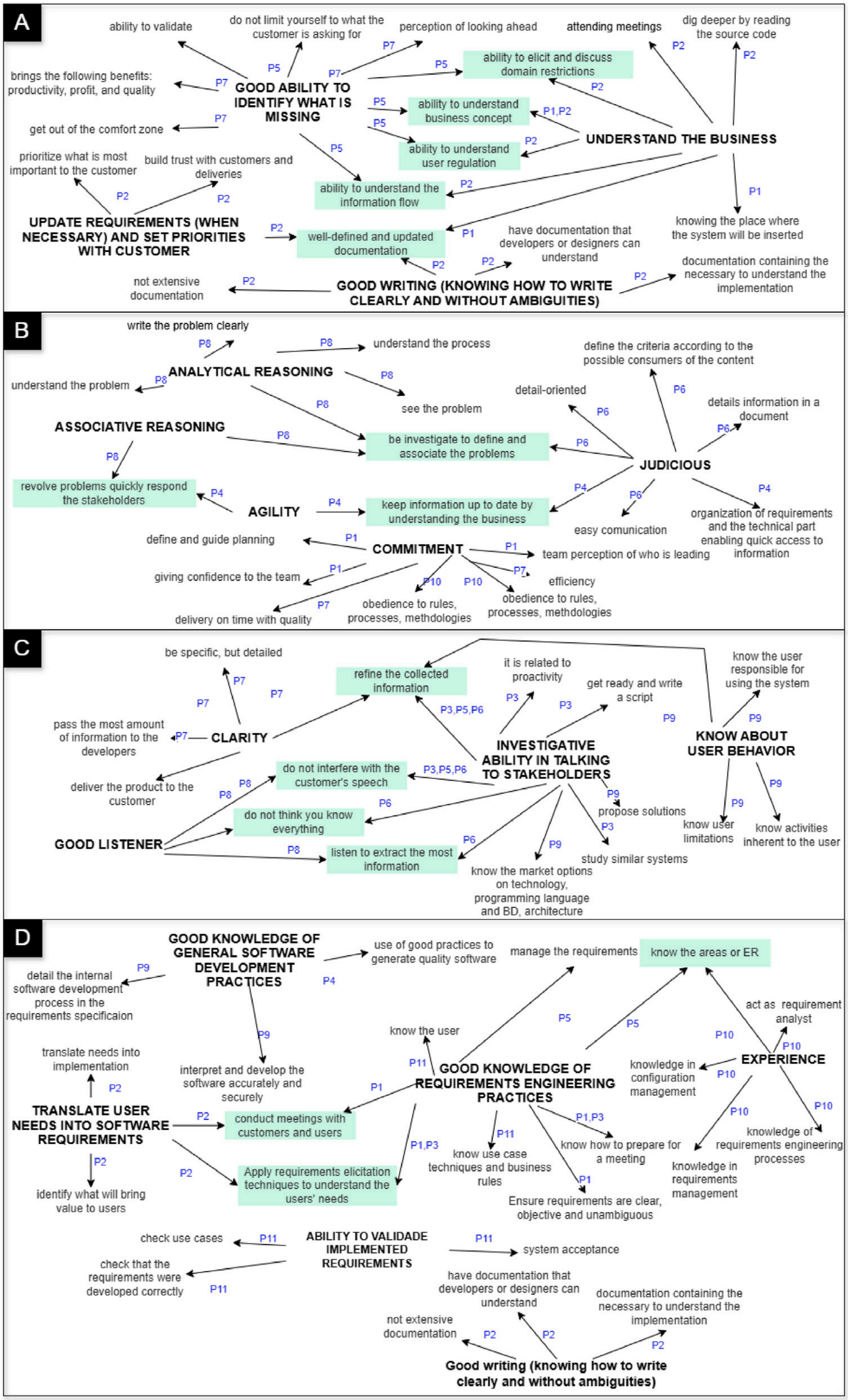
A
ability to validate
do not limit yourself to what the customer is asking for
perception of looking ahead
attending meetings
dig deeper by reading the source code
brings the following benefits: productivity, profit, and quality
GOOD ABILITY TO IDENTIFY WHAT IS MISSING
ability to elicit and discuss domain restrictions
ability to understand business concept
get out of the comfort zone
ability to understand user regulation
UNDERSTAND THE BUSINESS
prioritize what is most important to the customer
build trust with customers and deliveries
ability to understand the information flow
UPDATE REQUIREMENTS (WHEN NECESSARY) AND SET PRIORITIES WITH CUSTOMER
well-defined and updated documentation
have documentation that developers or designers can understand
knowing the place where the system will be inserted
not extensive documentation
GOOD WRITING (KNOWING HOW TO WRITE CLEARLY AND WITHOUT AMBIGUITIES)
documentation containing the necessary to understand the implementation
P1
P1,P2
P2
P5
P7
B
write the problem clearly
understand the process
ANALYTICAL REASONING
define the criteria according to the possible consumers of the content
understand the problem
see the problem
detail-oriented
details information in a document
ASSOCIATIVE REASONING
be investigate to define and associate the problems
JUDICIOUS
revolve problems quickly respond the stakeholders
AGILITY
keep information up to date by understanding the business
easy comunication
organization of requirements and the technical part enabling quick access to information
define and guide planning
COMMITMENT
team perception of who is leading
giving confidence to the team
efficiency
obedience to rules, processes, methdologies
delivery on time with quality
obedience to rules, processes, methdologies
P1
P4
P6
P7
P8
P10
C
be specific, but detailed
refine the collected information
it is related to proactivity
get ready and write a script
know the user responsible for using the system
pass the most amount of information to the developers
CLARITY
INVESTIGATIVE ABILITY IN TALKING TO STAKEHOLDERS
KNOW ABOUT USER BEHAVIOR
deliver the product to the customer
do not interfere with the customer's speech
propose solutions
do not think you know everything
know user limitations
know activities inherent to the user
GOOD LISTENER
listen to extract the most information
study similar systems
know the market options on technology, programming language and BD, architecture
P3
P3,P5,P6
P6
P7
P8
P9
D
GOOD KNOWLEDGE OF GENERAL SOFTWARE DEVELOPMENT PRACTICES
use of good practices to generate quality software
manage the requirements
know the areas or ER
detail the internal software development process in the requirements specificaion
act as requirement analyst
know the user
translate needs into implementation
interpret and develop the software accurately and securely
GOOD KNOWLEDGE OF REQUIREMENTS ENGINEERING PRACTICES
knowledge in configuration management
EXPERIENCE
TRANSLATE USER NEEDS INTO SOFTWARE REQUIREMENTS
conduct meetings with customers and users
know use case techniques and business rules
know how to prepare for a meeting
knowledge of requirements engineering processes
knowledge in requirements management
Apply requirements elicitation techniques to understand the users' needs
identify what will bring value to users
Ensure requirements are clear, objective and unambiguous
check use cases
ABILITY TO VALIDADE IMPLEMENTED REQUIREMENTS
system acceptance
check that the requirements were developed correctly
have documentation that developers or designers can understand
not extensive documentation
documentation containing the necessary to understand the implementation
Good writing (knowing how to write clearly and without ambiguities)
P1
P1,P3
P2
P4
P5
P9
P10
P11

Fig. 6. Map for (A) management, (B) personality, (C) social, and (D) technical categories.

improvements to achieve or improve these attributes in their projects. As a conceptual guide, the map can be employed to inform actions in response to perceived attributes, and as a comprehensive guide to evaluate strategies or improvements that can be achieved so that these attributes are increasingly present in our team. It facilitates the identification and recognition of attributes (represented in bold capital text) and concepts related to each attribute (represented in smaller, lowercase text).

5.2. Comparison to related work

Table 10 presents a comparison between our results, text books (columns in gray), and related work (columns in blue). Considering the attributes described in the text books (Pohl and Rupp, 2015; Häuß er et al., 2022), we can see that the common attributes are more related to human-aspects, such as *analytic thinking*, *responsibility*, *communication*, and *persuasiveness*. The other attributes found in our study reveal other abilities that requirements engineers have to pursue to develop their activities.

Concerning related work (columns in blue), the common attributes between a great requirements engineer and a great software engineer (Li et al., 2015) are related to soft skills (*agility*, *commitment*, *curiosity*, *good systematic view*, and *judicious*) and knowledge on processes, practices, and domain (*good knowledge of general software development practices*, *good knowledge of requirements engineering practices*, and *understand the business*). The other attributes found only for requirements engineers are associated with specific requirements activities such as *good listener*, *good writing (knowing how to write clearly and without ambiguities)*, and *translate user needs into software requirements*.

We also found a few common attributes with great open-source software maintainers, managers of software engineers, and leadership in agile software teams.

In summary, results confirmed our initial guess that attributes of great software practitioners might not be easily translated to the context of requirements engineering. Our results complement the set of information already reported in technical literature, providing specific attributes to requirements engineers.

5.3. Implications

The attributes we have identified and described may have wide-ranging implications for researchers, managers, novice requirements engineers, and educators. In the rest of this section, we discuss implications and opportunities to build upon our results.

5.3.1. For researchers

Researchers can consider the identified attributes in the development of new methods, strategies, and tools that help requirements engineers in carrying out their activities and, by making use of them, achieve the desirable attributes of a great requirements engineer.

Researchers might also investigate interventions that help achieve the attributes quickly and effectively.

5.3.2. For managers

The map can be used to support managers to more effectively communicate attributes of a great requirements engineer. As a conceptual guide, the proposed map along with the Sankey diagram facilitate a more effective identification, acknowledgment and relationship of attributes of great requirements engineers. Software practitioners can also use them to become better requirements engineers. For instance, managers can identify weak points in their teams and define strategies to reduce or even eliminate those weaknesses. If a manager identifies that the attribute *good knowledge of requirements engineering practices* is not ideal in her/his team, analyzing the relationships of this attribute, he/she can realize that by pursuing the concepts managing the requirements and knowing the requirements engineering areas he/she will be able to improve her/his team's skills.

5.3.3. For novice requirements engineers

It is a common issue to see new requirements engineers unsure of how to become great requirements engineers. The results reported in this work list a series of attributes that they might pursue. Moreover, our results provide a set of strategies that can be employed to this end. Improvements might come, for instance, from participating in training, acquiring domain knowledge, or talking to all stakeholders.

Our findings may also help new requirements engineers better present themselves in job searching. Novice engineers can consider demonstrating to employers that they have or can develop these attributes.

5.3.4. For educators

Our results raise questions about curriculum choices, teaching methods and learning objectives in software engineering education. Educators may consider adding courses on topics not found in their current curricula. For example, a specific course on requirements engineering could be necessary to make possible the teaching of requirements engineering practices.

Software engineering educators might also use our results to review their teaching method. Many attributes of great requirements engineers are soft skills. Teaching methods based on experiential learning practices could be useful to improve students' soft skills (O'Brien et al., 2021). In experiential learning, students actively engage in complex tasks that reflect the problems they likely encounter in the workplace. To bridge the academic and employment skills and knowledge they are developing through these tasks, students participate in reflective activities that help them articulate the relevance and implications of the experience for lifelong learning (O'Brien et al., 2021).

Lastly, educators might also explicitly discussing what students will not learn in school, allowing them to be aware of potential knowledge gaps, preparing them to search for opportunities outside of the academic setting (e.g. internships). For example, attributes like *investigative ability in talking to stakeholders* may not be reasonable to teach in an academic setting.

6. Threats to validity

As with any empirical study, there may be threats and/or limits to our methods and findings (Wohlin et al., 2012).

External validity: Our analysis comes wholly from 18 software practitioners from the software development industry from one country (Brazil). This makes it unlikely that our results are representative of the views of software managers and requirements engineers in general. To minimize this threat, we considered a variety of participants in terms of years of experience, company size, and number of companies worked. Besides, Brazil is a continental-sized country, the largest in South America, presenting enormous landscape, economic and cultural diversity. An argument can be made, however, that ecological validity (Andrade, 2018) of the work (i.e., the extent to which these findings approximate other real-world scenarios) does hold. Another threat that may affect external validity may be related to how many of these individuals have been exposed to truly great software engineers, instead of a number of more-or-less competent ones. To mitigate this threat, participants should justify why they consider the software engineers to be a great software engineer based on the attributes they define. In addition, we may have the fact that our domain is returning to the area of information systems. And to mitigate this threat we consider different domains (such as, commercial and financial).

Internal validity: Threats affecting the internal validity can arise from the questionnaire. As it was applied remotely, the participants could misinterpret the questions. To mitigate this threat, the questionnaire was submitted to two internal and one external validations, and a pilot study. Another threat is related to the term great requirements engineer used in the questions. As the participants could misinterpret this term, they could give invalid answers. To reduce this threat, we

Table 10
Comparison with related work.

	Attributes for a					
Great requirements engineer (our study)	Requirements engineer (Pohl and Rupp, 2015)	Requirements engineer (Häuß er et al., 2022)	Great software engineer (Li et al., 2015)	Great manager of software engineers (Kalliamvakou et al., 2019)	Great open-source software maintainer (Dias et al., 2021)	Great leadership in agile software teams (Gren and Ralph, 2022)
Ability to validate implemented requirements	–	–	–	–	–	–
Agility	–	–	Productive	–	–	–
Analytical reasoning	Analytic thinking	–	–	–	–	–
Antifragility (it is a concept that emphasizes the ability to evolve in the face of adverse situations)	–	–	–	–	Diligence	–
Associative reasoning	–	–	–	–	–	–
Clarity	–	–	–	–	–	–
Commitment	–	Responsibility	Hardworking	–	Responsibility	Step in, if needed
Curiosity	–	–	Curious	–	–	–
Experience	–	–	–	–	–	–
Good ability to identify what is missing	–	–	–	–	–	–
Good knowledge of general software development practices	–	–	Knowledgeable about engineering processes	Is technical	–	–
Good knowledge of requirements engineering practices	–	–	Knowledgeable about their technical domain	–	Technical excellence	–
Good listener	Communication skills	Communication	–	–	–	–
Good systematic view	–	–	Systematic	–	–	–
Good writing (knowing how to write clearly and without ambiguities)	Communication skills	Communication	–	–	–	–
Investigative ability in talking to stakeholders	Persuasiveness	–	–	–	Communication	–
Judicious	–	–	Attentive to details	–	Discipline	–
Knowledge about user behavior	–	–	–	–	–	–
Prior knowledge, even if basic, in some programming language	–	–	–	–	–	–
Translate user needs into software requirements	–	–	–	–	–	–
Understand the business	–	–	Knowledgeable about customers and business	–	Domain Experience	Understanding the company ecosystem
Update requirements (when necessary) and set priorities with customer	–	–	–	–	–	–

analyzed all answers given to SQ9 and concluded that no invalid answer was reported.

Conclusion validity: A threat arises from the qualitative analyses we carried out because they are a subjective task. To mitigate this threat, the analyses were carried out separately by two researchers and the consensus was performed by an experienced researcher. We also used the constant comparison technique when coding the interview transcripts, comparing our findings with previous ones as they emerged from the data analysis. Another threat affecting the conclusion validity is related to the participants' project role. Although we recognize the importance of requirements engineers in the context of this work, we did not survey only practitioners performing requirements tasks. Having a set of participants encompassing managerial tasks is also necessary because we can further understand the attributes of a great requirements engineer, as demonstrated in Section 4.2.1.

7. Concluding remarks

In this paper, we investigate the attributes of a great requirements engineer. We organize these attributes into a map that can support software professionals in seeking strategies to become better requirements engineers. Also, the map can guide new research efforts in a problem-driven way.

Our results also open many opportunities of future work. Examples of research gaps implications of the results for supporting requirements engineers, managers and educators are discussed in Section 5.3. Particularly, as next steps of this research, we intend to (i) conduct a rating survey with requirements engineers and project managers aiming at understanding how they prioritize the identified attributes, (ii) investigate undesirable attributes of a requirements engineer, and (iii) investigate teaching strategies that are suited to educate new requirements engineers considering both technical and soft skills.

CRediT authorship contribution statement

Larissa Barbosa: Writing – review & editing, Writing – original draft, Validation, Methodology, Data curation, Conceptualization. **Sávio Freire:** Writing – original draft, Methodology, Investigation, Formal analysis, Data curation. **Rita S.P. Maciel:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology. **Manoel Mendonça:** Writing – original draft, Supervision, Methodology. **Marcos Kalinowski:** Writing – original draft, Validation, Conceptualization. **Zadia Codabux:** Writing – original draft, Validation, Methodology, Conceptualization. **Rodrigo Spínola:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Savio Freire reports financial support was provided by Coordination of Higher Education Personnel Improvement. Larissa Barbosa reports financial support was provided by Coordination of Higher Education Personnel Improvement. Manoel Mendonca reports financial support was provided by National Council for Scientific and Technological Development.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank all software practitioners who participated in the experimental studies. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Finance Code 001 and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

References

Andrade, C., 2018. Internal, external, and ecological validity in research design, conduct, and evaluation. Indian J. Psychol. Med. 40 (5), 498–499. http://dx.doi.org/10.4103/IJPSYM.IJPSYM_334_18, PMID: 30275631.

Barbosa, L., Freire, S., Maciel, R.S.P., Mendonça, M., Kalinowski, M., Codabux, Z., Spínola, R., 2023. Complementary material. [Online]. Available: https://zenodo.org/records/13336394.

Ben-Akiva, M., Morikawa, T., Shiroishi, F., 1992. Analysis of the reliability of preference ranking data. J. Bus. Res. 24 (2), 149–164, [Online]. Available: https://www.sciencedirect.com/science/article/pii/014829639290058J.

Buckingham, M., 2005. What great managers do. IEEE Eng. Manag. Rev. 33 (2), 3.

Damian, D., Chisan, J., 2006. An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality, and risk management. IEEE Trans. Softw. Eng. 32 (7), 433–453.

Dias, E., Meirelles, P., Castor, F., Steinmacher, I., Wiese, I., Pinto, G., 2021. What makes a great maintainer of open source projects? In: 2021 IEEE/ACM 43rd International Conference on Software Engineering. ICSE, pp. 982–994.

Fernández, D.M., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., Conte, T., Christiansson, M.-T., Greer, D., Lassenius, C., Männistö, T., Nayabi, M., Oivo, M., Penzenstadler, B., Pfahl, D., Prikladnicki, R., Ruhe, G., Schekelmann, A., Sen, S., Spinola, R., Tuzcu, A., de la Vara, J.L., Wieringa, R., 2017. Naming the pain in requirements engineering. Empir. Softw. Eng. 22 (5), 2298–2338. http://dx.doi.org/10.1007/s10664-016-9451-7.

Gren, L., Ralph, P., 2022. What makes effective leadership in agile software development teams? In: Proceedings of the 44th International Conference on Software Engineering. ICSE '22, Association for Computing Machinery, New York, NY, USA, pp. 2402–2414. http://dx.doi.org/10.1145/3510003.3510100.

Häuß er, D., Lauenroth, K., van Loenhoud, H., Schwarz, A., Steiger, P., 2022. Handbook for the Advanced Level Requirements Elicitation. International Requirements Engineering Board.

Hemon, A., Lyonnet, B., Rowe, F., Fitzgerald, B., 2020. From agile to DevOps: Smart skills and collaborations. Inform. Syst. Front. 22 (4), 927–945. http://dx.doi.org/10.1007/s10796-019-09905-1.

Kalliamvakou, E., Bird, C., Zimmermann, T., Begel, A., DeLine, R., German, D.M., 2019. What makes a great manager of software engineers? IEEE Trans. Softw. Eng. 45 (1), 87–106.

Li, P.L., Ko, A.J., Zhu, J., 2015. What makes a great software engineer? In: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, vol. 1, pp. 700–710.

Lupton, R., Allwood, J., 2017. Hybrid sankey diagrams: Visual analysis of multidimensional data for understanding resource use. Resour. Conserv. Recycl. 124, 141–151, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921344917301167.

O'Brien, W., Doré, N., Campbell-Templeman, K., Lowcay, D., Derakhti, M., 2021. Living labs as an opportunity for experiential learning in building engineering education. Adv. Eng. Inform. 50, 101440, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034621001920.

Pohl, K., Rupp, C., 2015. Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam, Foundation Level, IREB Compliant. Rocky Nook.

Rivera-Ibarra, J.G., Rodríguez-Jacobo, J., Fernández-Zepeda, J.A., Serrano-Vargas, M.A., 2010. Competency framework for software engineers. In: 2010 23rd IEEE Conference on Software Engineering Education and Training. pp. 33–40.

Seaman, C., 1999. Qualitative methods in empirical studies of software engineering. IEEE Trans. Softw. Eng. 25 (4), 557–572.

Strauss, A.L., Corbin, J.M., 1998. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publications, Thousand Oaks, Calif, XIII, 312 s.

Wohlin, C., Runeson, P., Hst, M., Ohlsson, M.C., Regnell, B., Wessln, A., 2012. Experimentation in Software Engineering. Springer Publishing Company, Incorporated.

Larissa Barbosa is a Ph.D. student at Federal University of Bahia. Her research interest encompasses requirements engineering and technical debt. Contact her at larissa.barbosa11@gmail.com.

Sávio Freire is an Assistant Professor at the Federal Institute of Ceará, Morada Nova, Ceará, 62.940-000, Brazil. He is a researcher at the Technical Debt Research Team (http://www.tdresearchteam.com) at the Salvador University. His research interest encompasses technical debt and empirical software engineering. Contact him at savio.freire@ifce.edu.br.

Rita Suzana (rita.suzana@ufba.br) is an Associate Professor of computer science at the Federal University of Bahia (UFBA). Her research interests include Software Engineering, Information Systems, Collaborative Systems and Information Technology in Education.

Manoel Mendonç (manoel.mendonca@ufba.br) is an Associate Professor of computer science at the Federal University of Bahia (UFBA). He acted as the Founding Director of the Fraunhofer Center for Software and Systems Engineering at UFBA.

Marcos Kalinowski (kalinowski@inf.puc-rio.br) is an Associate Professor at the Department of Informatics at the Pontifical Catholic University of Rio de Janeiro (PUC-Rio). Co-founder and coordinator of the ExACTa initiative that helps partner companies in their digital transformation. More can be found at http://www.inf.puc-rio.br/~kalinowski.

Zadia Codabux (zadiacodabux@ieee.org) is an Assistant Professor in the Department of Computer Science at the University of Saskatchewan, Canada. Her research interests include technical debt, empirical software engineering, and software security. She completed her Ph.D. in Computer Science at Mississippi State University, USA and her MS in Computer Science at the University of Mauritius.

Rodrigo Spínola is an Associate Professor at Virginia Commonwealth University, Richmond, Virginia, 23284, United States, where he leads the Technical Debt Research Team (http://www.tdresearchteam.com). His research interests include technical debt and empirical software engineering. He holds a Ph.D. and MS in Computer Science and Systems Engineering from the Federal University of Rio de Janeiro. More can be found at http://www.rodrigospinola.com. Contact him at spinolaro@vcu.edu.