Toward Threshold Concepts for Teaching Requirements Engineering in Higher Education

Marian Biekart[1,*], Fabiano Dalpiaz[1]

[1]Utrecht University, Heidelberglaan 8, 3584 CS, Utrecht, Netherlands

Abstract

As the boundaries of the Requirements Engineering (RE) discipline keep expanding, deciding on which contents a course or module should include has become challenging. In absence of an agreed-upon syllabus for RE higher education, we take a step back and conduct a study that aims at identifying *threshold concepts*: a subset of the core RE concepts that, when mastered, are expected to let the learner make a leap in the understanding of the discipline. Through a two-phase mixed-methods design, consisting of individual semi-structured interviews followed by two focus groups to achieve consensus, we identified *two* threshold concepts as well as *seven* candidate ones. The two threshold concepts pertain to pillars of the RE discipline: understanding that RE is a co-creation process between analysts and stakeholders, and the importance of eliciting goals for uncovering the stakeholders' true needs.

Keywords

Requirements Engineering, Computing Education, Threshold Concepts

1. Introduction

The boundaries of the Requirements Engineering (RE) discipline keep expanding. First, the community has long acknowledged the influence from and the adaptation of techniques from various disciplines like social sciences [1]. Second, RE has been proposed for system types that go beyond software, including socio-technical [2], self-adaptive [3], cyber-physical [4], and artificial intelligence [5, 6]. Third, RE sources have expanded, moving from RE as specification document-centric (encoded in standards like IEEE's [7]), to a view where requirements originate from heterogeneous sources including agile artifacts [8] and online user feedback [9].

In this dynamic context [10], and given the wide range of existing viewpoints on the discipline, testified by the many textbooks such as [11, 12, 13], RE educators are confronted with the difficult selection of the contents to include in a given module or course, within the constraints given by the students' background, the surrounding curriculum, the course duration, etc.

While the situation for RE professionals training is somewhat more established and mature, also thanks to the existing syllabi defined by organizations such as the International Requirements Engineering Board (IREB), there is a lack of a standardized RE curriculum for higher

In: A. Hess, A. Susi, E. C. Groen, M. Ruiz, M. Abbas, F. B. Aydemir, M. Daneva, R. Guizzardi, J. Gulden, A. Herrmann, J. Horkoff, S. Kopczyńska, P. Mennig, M. Oriol Hilari, E. Paja, A. Perini, A. Rachmann, K. Schneider, L. Semini, P. Spoletini, A. Vogelsang. Joint Proceedings of REFSQ-2025 Workshops, Doctoral Symposium, Posters & Tools Track, and Education and Training Track. Co-located with REFSQ 2025. Barcelona, Spain, April 7, 2025.

*Corresponding author.
mj.biekart@gmail.com (M. Biekart); f.dalpiaz@uu.nl (F. Dalpiaz)
0000-0003-4480-3887 (F. Dalpiaz)

education, as demonstrated by the systematic literature review by Ouhbi and colleagues [14].

In this paper, we take a step toward this end by putting forward the following research question (**RQ**): *What are the threshold concepts [15] that, when mastered, allow a learner to progress to a deeper understanding of the RE discipline?*

We decided to investigate threshold concepts (TCs), rather than focusing on specific intended learning outcomes, topics, or techniques, because such 'big ideas' are expected to define the essential building blocks for learners to advance within a field. They are a subset of the *core concepts*, which are important but do not necessary lead to a learning leap.

The presented TCs originate from university-level RE teachers, who participated in two phases: (1) six teachers identified potential TCs, and (2) seven evaluated these concepts' transformative and troublesome characteristics. Employing a mixed-methods design, we used semi-structured interviews in the first, divergent phase and a modified Nominal Group Technique (NGT) for consensus-building in the second, convergent phase.

The rest of paper is structured as follows. We first introduce the notion of TCs in Section 2. Then, we outline our research method in Section 3. We review the identified TCs in Section 4, and finally present discussion and outlook in Section 5.

2. Background: Threshold Concepts

Meyer *et al.* [15] found that, in the field of economics, there are certain moments in the program where students are generally getting stuck. Yet, these troublesome points are often considered essential for a deeper understanding of the subject. They described these moments as "thresholds", because understanding them is like crossing a portal; once understood properly, they fundamentally change one's perception of the subject and unlock deeper understanding.

They posit that TCs exist in any discipline, just like there are core concepts. While both concept types are required to understand a subject, TCs lead to a qualitatively different view of the subject matter or a new level of understanding, whereas core concepts do not. TCs are a subset of core concepts that have transformative power [16]. To distinguish TCs from core concepts, Meyer *et al.* [15] identified five characteristics that TCs are *likely* to have: *transformative*; probably *irreversible*; *integrative*; often *bounded*; and potentially (and possibly inherently) *troublesome*.

There is still debate on whether all these characteristics are necessary for a core concept to be a threshold one. However, the literature advocates that two of them—transformative and troublesome—are the most influential in determining the learning success or failure and are the easiest to measure [17, 18]. We focus on those two in this research (definitions from [15]):

- *Transformative:* Once understood, a threshold concept causes the learner to experience a shift in perception or way of understanding a subject;
- *Troublesome:* A threshold concept is generally hard to grasp because it involves troublesome knowledge, which may be conceptually difficult, alien, tacit, inert, or ritual.

3. Research Method Outline

As explained in the introduction, informed by the review of Correia *et al.* [18], we designed our study as a two-stage approach with inputs from university-level RE teachers. We first followed a divergent approach (Section 3.1), where we conducted six semi-structured interviews with teachers to propose potential TCs. Then, we used a modified NGT to evaluate these concepts based on the transformative and troublesome characteristics (Section 3.2).

3.1. Phase 1 (divergent): Semi-structured interviews

The first phase aimed to elicit individual experiences and insights into potential TCs for RE, while avoiding the risk of group dynamics such as dominant voices limiting others' input [19]. We identified participants based on their profiles with the requirement of having at least three years of teaching experience in RE at the university level. Thus, they were selected for their suitability and relevance to this study. In total, six expert teachers, two women and four men, participated, having affiliations with computer science and software engineering departments of universities in Germany (3), Italy (1), Sweden (1) and Switzerland (1). Their RE teaching experience ranged from 6-7 years to almost 30 years. To guide our interviews, we relied on the Content Representation form by Loughran *et al.* [20], which we modified to fit our specific needs by combining elements from Loughran's original eight questions with questions from the transactional curriculum inquiry [21]. Table 1 shows the interview protocol we followed. For accessibility reasons, we used the term *big ideas* rather than TCs. We acknowledge that the two are not identical (although closely related); however, not all teachers are familiar with the TC theory, so using the term *big ideas* encourages them to share what they perceive as important in RE without worrying about the precise definition. This approach allows teachers to express ideas more freely, without the need to evaluate whether these concepts meet the specific characteristics of TCs.

Table 1
The semi-structured interview protocol we used in Phase 1.

0. What do you consider to be the *Big Ideas* in teaching Requirements Engineering?

For each Big Idea:

1. Why is it important for students to know this idea?
2. What do you intend the students to know about this idea?
3. To what extent is mastery of this idea troublesome?
    3.1 What misunderstandings do students characteristically exhibit?
4. In what way can mastery change the learner's perception of the subject?
5. How do you typically teach this idea (and what are the particular reasons for using these to engage with this idea)?

3.2. Phase 2 (convergent): Nominal Group Technique

To ensure the educational value of the identified concepts and avoid the risk of identifying a great number of TCs with little agreement, we followed the reasoning of Barradell [17] and used the NGT as consensus-building approach. Given the participants' diverse locations, we conducted the NGT online, using Microsoft Teams as online videoconferencing tool. To evaluate the proposed TCs and gather diverse perspectives, we recruited a new group of expert RE teachers. In this phase, seven teachers, two women and five men, participated, having affiliations with computer science and software engineering departments of universities in the Netherlands (3), United Kingdom (1), Italy (1), Sweden (1) and Germany (1). To maximize the contribution of each participant [22], they were divided into two smaller groups, one with four participants and the other with three. Each group followed a structured protocol, based on an adapted version of NGT, consisting of the following steps:

1. Introduction, to inform the participants on study objectives and session protocol.
2. Clarification, to reach a shared understanding of each of the proposed TCs.
3. Explanation, to outline the TC theory, with emphasis on the two criteria.
4. Voting, to collect each participant's individual perception of the *transformative* nature of each proposed concept by rating it on a 4-point Likert scale (1 = disagree, 4 = agree).
5. Discussion on the aggregate scores, with focus on highly rated concepts, concepts with inconsistent ratings, any concepts perceived as over- or under-rated.
6. Voting, to collect each participant's individual perception of how *troublesome* each concept is, using the same rating process of step 4.

We considered concepts as meeting the transformative or troublesome characteristics if and only if they had only '(somewhat) agree' ratings, with at most one 'somewhat disagree' rating.

4. Identified Threshold Concepts

The first phase led to eighteen candidate TCs. Out of these, after the discussion and voting in the second phase, *nine* concepts were retained. For two concepts, the participants agreed on both the transformative and troublesome characteristics (see Section 4.1); for the remaining seven, the participants agreed only on transformativeness (see Section 4.2). The eighteen concepts and their ratings are in our online appendix [23].

4.1. The two pillars: transformative and troublesome

> **TC1**: *Stakeholders' desires and needs are not merely discovered, but actively shaped into precise requirements through a co-creation process.*

Our participants found that students often believe that requirements are in the stakeholders' heads, and just need to be collected by asking them what they want. However, the realization that RE is not a passive task, but an active, collaborative process can change the students' perception. Both stakeholders and requirements engineer(s) work together to shape the requirements, which may lead to different outcomes compared to a requirements engineer acting as a passive recorder. As one participant stated, "It's not just about asking stakeholders and recording their answers; it's about actively shaping the requirements through collaboration." This aligns with research that shows the importance of user involvement [24], participation of user crowds [9, 25], and the conduction of group workshops that foster creativity [26].

TC2: *Understanding the goals behind requirements is essential for uncovering true needs and exploring alternatives.*

Many students take stakeholder statements at face value and act like record keepers/accountants rather than critical thinkers. Understanding that the stated requirements (what the stakeholders explicitly ask for) often reflect pre-conceived solutions rather than real needs (the underlying problem or purpose that needs to be addressed) can help them realize that RE is not straightforward. As one participant put it, "They should not just believe that they are acting as an accountant [...] They have to question it (a stated requirement). They have to question whether it is well expressed or actually, there's something different behind it." By asking *why* questions and digging deeper, students can uncover the actual problem, explore alternative solutions, and often find that what was provided as a requirement by the stakeholder might not be necessary. TC2 is a clear sign that educators acknowledge the importance of focusing on the problem [27] and thereby exploring goals [13, 28, 29] to analyze the *why* dimension.

4.2. The seven candidates: transformative, not troublesome

In addition to the two TCs listed in the previous section, seven other ideas met our criteria for the transformative characteristic, but were not considered troublesome. We discuss these possible candidates in thematic clusters. Further investigation is necessary to determine if these qualify as TCs for the RE discipline.

4.2.1. Context dependence

TC3: *Systems are embedded in context and cannot be developed in isolation.*
TC4: *Requirements and their context, including systems and people, are in constant evolution, thereby requiring iterative approaches to RE.*
TC5: *There is no universal RE method: flexibility and adaptation are required to cope with context dependency.*

TC3 expresses the necessity of looking beyond the software to-be and adopting a perspective in which RE focuses on the context/environment where the software will be placed [30], which includes legacy systems, and relies on domain properties and assumptions [13, 27]. TC4 highlights an additional facet: requirements change (evolve) over time [31], and this entails that RE is not a waterfall discipline; iterative methods are a necessity [32]. The combination of TC3 and TC4 inevitably leads to TC5: the variety of RE methods and techniques is justified by the need to select an approach that caters for the contextual aspects. Research argues that this applies also when choosing the technique(s) to employ for given RE activities, such as elicitation [33].

4.3. Effort in RE? The risk-value trade-off

TC6: *The effort put into RE activities should strike a balance between the expected value for conducting the activity and the potential risk arising when not doing the activity.*
TC7: *Although obtaining a complete and unambiguous requirements specification is an illusion, it is important to achieve a sufficient level of clarity and precision.*

While RE is often been argued as a success factor for software projects [34], the return-on-investment is hard to measure. With TC6, RE educators acknowledge the inherent trade-off between potential risk and value; this is in line with IREB's definition of RE [35], which states how RE is about "[...] minimizing the risk of delivering a system that does not meet these desires and needs." TC7 exemplifies a typical RE activity where excessive effort may be harmful or not justified: requirements specification. The participants acknowledge that although a perfect specification (unambiguous, complete, ...) is a utopia, some effort should be placed to support communication between analysts and stakeholders. This is in line, for instance, with the distinction between nocuous and innocuous ambiguity in RE [36].

4.4. Representing requirements

TC8: *For large-scale systems, requirements must be systematically organized and structured in order to support their use by both the development team and stakeholders.*
TC9: *Requirements are not always explicitly labeled as such, and they do not necessarily have a text-only representation.*

The participants shared different perspectives regarding how requirements are and should be represented. TC8 stresses that students often see requirements as simple lists. However, they must learn that documents must be thoughtfully organized. This prepares them for industry, where effective requirements management is crucial for complex systems [11]. In addition, TC9 refers to an often overlooked aspect: requirements are not only represented in specification documents with a 'requirement' label. Researchers have acknowledged this with studies on the role of requirements beyond specification documents, including law and regulatory documents [37], pre-tender phases [38], agile development [8, 39], online user feedback [40, 9], and vision videos [41].

5. Discussion and Outlook

As the two pillars TC1 and TC2 (Section 4.1) exhibit the key transformative and troublesome characteristics of a TC, they represent "jewels in the curriculum." Educators teaching RE in higher education should prioritize these concepts as explicit learning objectives. Future research should investigate how to effectively guide students across these thresholds through appropriate topics, didactics, and assessments. Since we could not draw a conclusion on the candidate threshold concepts TC3–TC9, these should be investigated more in depth with additional educators to determine whether they can become pillars too.

Furthermore, while involving participants from various universities helps reduce bias associated with a single institution, our sample size may be insufficient to represent the broader population. Expanding the scope of the research beyond Europe can provide insight into the extent to which the identified concepts are universally recognized as TCs or whether they are specific to the study's context. Similarly, investigating the perspectives of other stakeholders, including students and practitioners, could reveal whether additional candidates exist, thereby contributing to a more comprehensive understanding.

Acknowledgments

We would like to thank all the RE educators who participated in our study by dedicating their precious time and by sharing their ideas with us.

References

[1] J. A. Goguen, Requirements engineering as the reconciliation of social and technical issues, in: Requirements Engineering: Social and Technical Issues, Academic Press Professional, Inc., 1994, pp. 165–199.

[2] G. Baxter, I. Sommerville, Socio-technical systems: From design methods to systems engineering, Interacting with Computers 23 (2011) 4–17.

[3] P. Sawyer, N. Bencomo, J. Whittle, E. Letier, A. Finkelstein, Requirements-aware systems: A research agenda for RE for self-adaptive systems, in: Proc. of the IEEE International Requirements Engineering Conference, 2010, pp. 95–103.

[4] P. Loucopoulos, E. Kavakli, N. Chechina, Requirements engineering for cyber physical production systems, in: Proc. of the International Conference on Advanced Information Systems Engineering, Springer, 2019, pp. 276–291.

[5] K. Ahmad, M. Abdelrazek, C. Arora, M. Bano, J. Grundy, Requirements engineering for artificial intelligence systems: A systematic mapping study, Information and Software Technology 158 (2023) 107176.

[6] F. Dalpiaz, N. Niu, Requirements engineering in the days of artificial intelligence, IEEE Software 37 (2020) 7–10.

[7] Iso/iec/ieee international standard - systems and software engineering – life cycle processes – requirements engineering, ISO/IEC/IEEE 29148:2018(E) (2018) 1–104. doi:10.1109/IEEESTD.2018.8559686.

[8] E.-M. Schön, J. Thomaschewski, M. J. Escalona, Agile requirements engineering: A systematic literature review, Computer Standards & Interfaces 49 (2017) 79–91.

[9] E. C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, et al., The crowd in requirements engineering: The landscape and challenges, IEEE Software 34 (2017) 44–52.

[10] M. Glinz, The challenge(s) of teaching requirements engineering, 2021. Keynote Presentation at REFSQ 2021. Available for download at: https://2021.refsq.org/details/refsq-2021-papers/3/The-Challenge-s-of-Teaching-Requirements-Engineering.

[11] S. Robertson, J. Robertson, Mastering the requirements process: Getting requirements right, Addison-Wesley, 2012.

[12] K. Pohl, Requirements Engineering: Fundamentals, Principles, and Techniques, Springer, 2010.

[13] A. van Lamsweerde, Requirements engineering: From system goals to UML models to software specifications, John Wiley & Sons, Ltd, 2009.

[14] S. Ouhbi, A. Idri, J. L. Fernández-Alemán, A. Toval, Requirements engineering education: A systematic mapping study, Requirements Engineering 20 (2015) 119–138.

[15] J. Meyer, R. Land, Threshold concepts and troublesome knowledge: Linkages to ways

of thinking and practising within the disciplines, in: Improving Student Learning – Ten Years On, Oxford Centre for Staff & Learning Development, 2003.
[16] A. Eckerdal, R. McCartney, J. E. Moström, M. Ratcliffe, K. Sanders, C. Zander, Putting threshold concepts into context in computer science education, ACM SIGCSE Bulletin 38 (2006) 103–107.
[17] S. Barradell, The identification of threshold concepts: A review of theoretical complexities and methodological challenges, Higher Education 65 (2013) 265–276.
[18] P. R. Correia, I. A. Soida, I. de Souza, M. C. Lima, Uncovering challenges and pitfalls in identifying threshold concepts: A comprehensive review, Knowledge 4 (2024) 27–50.
[19] D. Shinners-Kennedy, S. A. Fincher, Identifying threshold concepts: From dead end to a new direction, in: Proc. of the International ACM Conference on International Computing Education Research, 2013, pp. 9–18.
[20] J. Loughran, P. Milroy, A. Berry, R. Gunstone, P. Mulhall, Documenting science teachers' pedagogical content knowledge through PaP-eRs, Research in Science Education 31 (2001) 289–307.
[21] G. Cousin, Researching learning in higher education: An introduction to contemporary methods and approaches, Taylor & Francis Routledge, 2009.
[22] F. Khurshid, E. O'Connor, R. Thompson, I. Hegazi, Twelve tips for adopting the virtual nominal group technique (vNGT) in medical education research, MedEdPublish 13 (2023).
[23] M. Biekart, F. Dalpiaz, Online Appendix of the paper "Toward Threshold Concepts for Teaching Requirements Engineering in Higher Education", 2025. URL: https://doi.org/10.5281/zenodo.14975499.
[24] M. Bano, D. Zowghi, Users' involvement in requirements engineering and system success, in: Proc. of the International Workshop on Empirical Requirements Engineering, IEEE, 2013, pp. 24–31.
[25] R. Snijders, F. Dalpiaz, S. Brinkkemper, M. Hosseini, R. Ali, A. Özüm, REfine: A gamified platform for participatory requirements engineering, in: Proc. of the International Workshop on Crowd-Based Requirements Engineering, IEEE, 2015.
[26] N. Maiden, A. Gizikis, S. Robertson, Provoking creativity: Imagine what your requirements could be like, IEEE Software 21 (2004) 68–75.
[27] P. Zave, M. Jackson, Four dark corners of requirements engineering, ACM Transactions on Software Engineering and Methodology 6 (1997) 1–30.
[28] A. Van Lamsweerde, E. Letier, From object orientation to goal orientation: A paradigm shift for requirements engineering, in: Proc. of the International Workshop on Radical Innovations of Software and Systems Engineering in the Future, Springer, 2002, pp. 325–340.
[29] E. Yu, J. Mylopoulos, Why goal-oriented requirements engineering, in: Proc. of the International Workshop on Requirements Engineering: Foundations of Software Quality, 1998, pp. 15–22.
[30] M. Jackson, Problem Frames: Analyzing and structuring software development problems, Addison-Wesley Longman Publishing Co., Inc., 2000.
[31] S. D. Harker, K. D. Eason, J. E. Dobson, The change and evolution of requirements as a challenge to the practice of software engineering, in: Proc. of the IEEE International Symposium on Requirements Engineering, IEEE, 1993, pp. 266–272.
[32] N. Ernst, A. Borgida, I. J. Jureta, J. Mylopoulos, An overview of requirements evolution,

Evolving Software Systems (2014) 3–32.
[33] D. Carrizo, O. Dieste, N. Juristo, Systematizing requirements elicitation technique selection, Information and Software Technology 56 (2014) 644–669.
[34] H. F. Hofmann, F. Lehner, Requirements engineering as a success factor in software projects, IEEE Software 18 (2001) 58.
[35] M. Glinz, Requirements Engineering Glossary, v 2.1.0, 2024. International Requirements Engineering Board.
[36] F. Chantree, B. Nuseibeh, A. De Roeck, A. Willis, Identifying nocuous ambiguities in natural language requirements, in: Proc. of the IEEE International Requirements Engineering Conference, IEEE, 2006, pp. 59–68.
[37] T. Breaux, A. Antón, Analyzing regulatory rules for privacy and security requirements, IEEE Transactions on Software Engineering 34 (2008) 5–20.
[38] G. Brataas, G. K. Hanssen, X. Qiu, L. S. Græslie, Requirements engineering in the market dialogue phase of public procurement: A case study of an innovation partnership for medical technology, in: Proc. of the International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2022, pp. 159–174.
[39] G. Lucassen, F. Dalpiaz, J. M. E. van der Werf, S. Brinkkemper, Improving agile requirements: The quality user story framework and tool, Requirements engineering 21 (2016) 383–403.
[40] W. Maalej, M. Nayebi, T. Johann, G. Ruhe, Toward data-driven requirements engineering, IEEE Software 33 (2015) 48–54.
[41] K. Schneider, M. Busch, O. Karras, M. Schrapel, M. Rohs, Refining vision videos, in: Proc. of the International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2019, pp. 135–150.