A Gamified Tutorial for Learning about Security Requirements Engineering

Daniel Alami
International Business Machines S.A.
Madrid, Spain
daniel.alami.cabezas@es.ibm.com

Fabiano Dalpiaz
Dept. of Information and Computing Sciences
Utrecht University
Utrecht, the Netherlands
f.dalpiaz@uu.nl

***Abstract*—Thanks to the advent of interactive technologies, education institutions are looking for innovative teaching methods to increase the engagement and reach of students. Besides the uprise of MOOCs, gamification has been shown to produce positive results when it comes to increasing people's engagement and interest in conducting tasks. Unfortunately, the application and benefits of these technologies in teaching requirements engineering remain largely unexplored. In this paper we introduce the STS-Tooltorial, an interactive gamified platform that executes within a security requirements modeling tool and helps learners apprehend the STS-ml language and basic notions about security requirements. We present the design principles of our functional prototype: its educational content and the embedded game elements. Furthermore, we report on an early evaluation with IT professionals and postgraduate information science students focused on the platform's effectiveness and usability.**

***Index Terms*—gamification, security requirements, requirements engineering, interactive tutorial, STS-ml.**

I. Introduction

Requirements Engineering (RE) is a crucial activity for creating high-quality software and a vital component of project success [1]. Among others, security is a key quality requirement of software systems. Failing to consider security requirements early in the development process increases the chances of introducing security breaches. Unfortunately, security is often an afterthought in software engineering [2], [3].

Several studies have highlighted the importance of including security since the early stages of system development [4], [5]. These efforts fall under the domain of Security RE (SRE), which emerged as a response to the monetary expenditures associated with (bad) security [6].

The quality of the SRE process is key to minimize the likelihood of introducing vulnerabilities. Many SRE approaches advocate the use of modeling languages that, based on conceptual modeling foundations [7], create models of the system to precisely document and analyze security requirements together with design requirements [8], [9]. The challenge for modelers is to adequately capture security requirements into the models.

The problem we tackle in this paper is that of "*How to effectively teach model-driven security RE*?" Mastering model-driven SRE requires the skills of developing, manipulating, and understanding models. Exercising these skills depends on both analytical capabilities and, in line with modern educational theories [10], an effective educational process.

Teaching conceptual modeling is not easy [11]. The quality of a conceptual model depends on both the modeler's understanding of the modeling language (rules, semantics and constructs) and her knowledge about the modeled domain [12].

In this paper, we build on modern education theories and we propose the use of an interactive gamified tutorial that is embedded within a security requirements modeling tool. We do so with the aim to foster self-learning of the basic concepts of SRE without relying on traditional (passive) lecturing methods that are proven to be ineffective for today's learners [13]. In particular, the paper makes the following contributions:

- We define a framework for building interactive tutorials for conceptual modeling languages. A key component of our framework is gamification [14], which we include to foster learner engagement by raising enjoyment and active participation (Sec. II).
- We describe the design of the curriculum and gamified experience of our STS-Tooltorial, which instantiates our framework for SRE and for the STS-ml goal-oriented language [9], [15] in particular (Sec. III).
- We report on preliminary results on the effectiveness of our approach for learning the basics of STS-ml and SRE, based on two studies with small groups of students and industry professionals (Sec. IV).

We conclude the paper with a discussion of our approach and by sketching future directions (Sec. V).

II. A Framework for Building Interactive Tutorials for Conceptual Modeling Languages

Based on our study of the literature in innovative teaching methods, interactive tutorial design, and gamification of learning, we distill guidelines for building interactive tutorials for conceptual modeling languages. We considered peer-reviewed publications and gray literature, while we excluded studies that presented empirically unverified guidelines or best practices were excluded, for we wanted the STS-Tooltorial to leverage effective and empirical guidelines.

A. *Innovative Teaching Methods*

We focused our literature study on papers published in venues where software and requirements engineering researchers gather to share perspectives on education. Venues like the Conference on Software Engineering Education and

2332-6441/17 $31.00 © 2017 IEEE
DOI 10.1109/RE.2017.67

Training (CSEE&T), the ICSE Software Engineering Education and Training Track (SEET), and the workshop on Requirements Engineering Education and Training (REET) have called for the cooperation among multidisciplinary fields.

Our inquiry evidenced, over the last decade, a growing interest in the application of innovative learning methods, techniques, and tools for RE and SE curricula. Innovative methods for enriching classroom training include improvisation theater [16], role playing [17], [18], project simulations [19], and case-based courses with either virtual [20] or real stakeholders [21].

Interestingly, we found that in almost all cases, the material taught with these approaches is predominantly about the teaching of high-level, soft skills. Only a few approaches (e.g., Zowghi *et al.* [17]) dive deeper into teaching specific methods, techniques or tools.

Therefore, we observe that room for improvement exists for introducing a novel teaching approaches that go deep into teaching procedural knowledge about a method or technique.

B. Design of Interactive Tutorials

The design of interactive tutorials requires learner-centric approaches that differ from the teacher-centric methods often employed in traditional education. Our investigation revealed a solid baseline, including the principles and implications for the design of learning systems of Park and Hannafin [22], the principles and heuristics for minimalist instruction design [23], and the rich literature on computer tutoring.

Recent research points to (1) the use of video tutorials as a prevalent source of information for users, for which comprehensive guidelines exist [24], and (2) the use of contextual assistance for enhancing understanding, quality, and flow [25].

We assembled a collection of 33 guidelines for designing an interactive tutorial, which we use in Sec. III to inform the design of our STS-Tooltorial. An excerpt of these guidelines is reproduced in Table 1 (the complete list with references is available in an online appendix[1]).

TABLE I
EXCERPT OF GUIDELINES FOR DESIGNING INTERACTIVE TUTORIALS

ID	Guideline
1	Provide instructions in the problem-solving context, anchoring the tool in the task domain.
2	Promote an abstract understanding of the problem-solving context.
3	Minimize working memory load and complement cognitive processes by structuring presentations and interactions.
4	Provide on-the-spot, immediate error information that supports detection, diagnosis, and recovery.
5	Facilitate successive approximations to the target skills.
6	Provide an immediate opportunity to act.
7	Organize lesson segments into internally consistent, meaningful, and self-contained units.
8	Differentiate important information through cosmetic amplification, repetition, and recasting to direct learners' attention.
9	Use videos that: 1) provide procedural or instructional information rather than conceptual information, 2) keep segments short, 3) ensure tasks clarity, and 4) coordinate demonstrations with text.

[1] www.staff.science.uu.nl/~dalpi001/appendix-sts-tooltorial.pdf

C. Gamification of Learning

Effectively designing a gamified tutorial requires profound knowledge of the state of the art on gamification elements and best practices for gamifying learning experiences.

We studied both white and gray literature about game elements. The rationale for covering both was that key success factors are missing in most academic work, which seems to mainly rely on motivational affordances of an extrinsic nature. In contrast, industry gamification experts such as Chou [26] highlight the importance of intrinsic motivation (this is also confirmed by recent studies by academics such as Deterding [14]).

We found 17 game elements consistently reported in industry and academia, including point system, badge, leaderboard, level/mission, progress bar, story/narrative, avatar, easter egg, time restriction, onboarding, game master, etc. After studying those game elements and their associated motivational affordances, we distilled a list of best practices for gamifying learning experiences (see Table II).

TABLE II
BEST PRACTICES FOR GAMIFYING LEARNING EXPERIENCES

Guideline	Description	Refs
Freedom of choice	Allow for freedom of choice. E.g., learners should be able to choose the order or speed of the challenges to be completed, or what goals to pursue. At a minimum, give the feeling of freedom.	[26]–[29]
Freedom to fail	Adopt the freedom to fail principle: poor task performance should not incur in penalties. E.g., learners should be able to retake quizzes.	[27]–[29]
Baby steps	Divide and present the educational content in small pieces of coherent information.	[30]
I know this!	Conduct an evaluation step (e.g., exercises, quizzes) after presenting educational content.	[30]
Together is better	Make the system social to motivate students via peer pressure or comparison with other students.	[27]–[30]
Boost it!	Give bonuses after the completion of hard tasks.	[28]–[30]
The good Samaritan	Compensate students not only for academic achievement but also prosocial behavior	[27]–[30]
Crystal clear	Offer immediate feedback and inform learners of their progression within the tutorial (e.g., progression bars. Frequent and immediate feedback leads to greater learning effectiveness and engagement.	[29]–[31]
Ethical designer	Design with ethics in mind: e.g., full transparency and opt-in principles.	[14], [26]
Know your users	Target population must be studied, regardless of whether gamification is considered or not	[28]
The road ain't easy	Communicate that the training will be challenging	[26]
Location, location	The location in which users engage with the application matters	[26]

Taken together, an understanding of 1) curricula design, 2) guidelines for designing interactive tutorials, 3) game elements and 4) principles for gamifying learning experiences constitute the framework that we used for designing the STS-Tooltorial.

III. STS-Tooltorial: A Gamified Tutorial for Teaching Security Requirements Engineering

We explored how to effectively teach a particular SRE method through an interactive tutorial: the Socio-Technical Security (STS) method, which is a goal-oriented and model-driven approach [9], [15]. Our choice is due to two main reasons: 1) STS is a socio-technical approach that considers security issues both from a technical and organizational/social perspective; and 2) the availability of a robust security requirements modeling tool: STS-Tool [32] [2].

Our interactive platform, called STS-Tooltorial, introduces learners to basic concepts of socio-technical systems and security requirements engineering to then delve into the STS method and modeling language. The platform is embedded in STS-Tool, thereby offering a first-of-a-kind experience for learners as they can learn at their own pace following a hands-on and highly interactive approach with short feedback cycles.

At the time of this writing, the tool covers a subset of the STS method that can be completed at once within 90 minutes. However, we are planning to release further learning modules to cover the entire spectrum of the STS method.

A. Curriculum design

To extract educational objectives, we supplemented desktop research with semi-structured interviews with seven leading SRE experts with background in teaching. Our goal was to identify a list of intended learning outcomes for a hypothetical two-hour classroom-based tutorial covering concepts of SRE and conceptual modeling. The final list was the following:

- *ILO1*: Identify the different activities and deliverables of security requirements engineering.
- *ILO2*: Recognize the modeling rules and elements as well as their meaning (in an existing model).
- *ILO3*: Choose the most appropriate set of elements for representing security-related aspects.
- *ILO4*: Express interest in pursuing further learning in the field of (socio-technical) SRE.

The interviews also provided useful information that helps complement the general guidelines on instructional design presented in the previous section, based on best practices and heuristics that academics use to teach SRE and modeling, regardless of the delivery method:

- *S1*: Modeling is an iterative process: models are constructed and refined in a stepwise fashion.
- *S2*: It is important to consider the attacker perspective.
- *S3*: It is key to stress the economic impact of security issues to raise awareness.
- *S4*: Distinguishing between document and information is important in socio-technical systems design.
- *S5*: Security is a multi-level concept (physical, network, and social). The social aspect is too often under-researched, under-focused, and under-prioritized.
- *S6*: Teaching through exercises, using a hands-on case-based approach.
- *S7*: Including a hands-on task only if the learner has previous experience.
- *S8*: Including a reflection phase after teaching.
- *S9*: Giving feedback after completing a modeling task.
- *S10*: Having students review each other's work through peer inspection.
- *S11*: It is important to capture the rationale of models.
- *S12*: SRE should not be seen as a separate activity. The task of analyzing a system and creating a good set of security requirements should be done while conducting other software development activities.

Note that while the heuristics focus on security requirements, some of them could potentially be generalized to general-purpose requirements. Therefore, the approach reported here may apply to modeling requirements in general. However, further research is necessary to determine to what extent the approach is generalizable.

Lastly, the STS-Tooltorial is aimed at novice modelers: it assumes learners have basic knowledge about conceptual modeling (e.g., UML) and information security (e.g., confidentiality, integrity, etc.). Our sample of students typically acquired these basic notions in a Bachelor's course.

B. Tutorial design

The general guidelines and the insight of SRE experts set the foundation for the design of the interactive platform. The STS-Tooltorial is a Web-based learning management platform based on Wordpress that can be embedded in the STS modeling tool. The delivery method consists of bite-sized lessons (*Levels*) lasting between 1 and 5 minutes where learners are progressively introduced to SRE, socio-technical systems, and the STS method and modeling language at increasing levels of difficulty to keep them in a *state of flow* [33]. The educational material is presented in the form of videos and text. In addition, the lessons contain *Quizzes*, which may consist of multiple-choice, true/false, or what we refer to as "model interaction" questions. The latter feature provides learners with questions about the content they just learned that prompt them to interact with the modeling tool, drawing diagrams and getting automated feedback on their correctness (see Fig. 1).

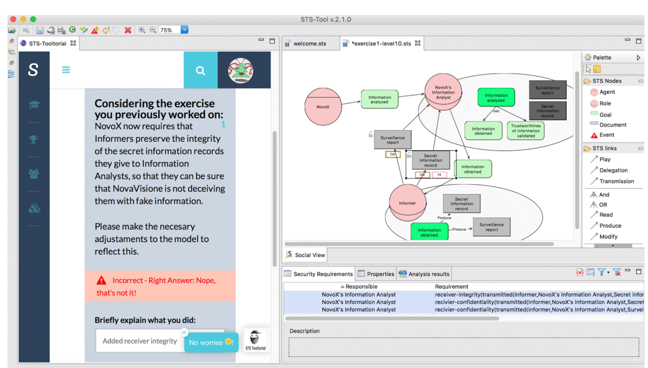

Fig. 1. Screenshot of the STS-Tooltorial (left) embedded in the STS-Tool

[2]http://www.sts-tool.eu/

C. Game elements

Aside from *Levels* and *Quizzes*, the use of other game-like elements is pervasive throughout the STS-Tooltorial, as they support the different learning activities and tap into both intrinsic and extrinsic motivators. Table III shows a comprehensive list of the game elements used in the interactive tutorial (the theoretical underpinnings and references are available in our online appendix). In the following, we present the storyline and how game elements are embedded therein.

TABLE III
SUMMARY OF THE IMPLEMENTED GAME ELEMENTS

Element	Description	Expected motivational affordances
Points	Points are granted to reward users upon task completion	Feedback, competition, sense of achievement, and positive emotions
Rank	Visualization of achievements to give a sense of progression	Progression, instruction, reputation, and group identification
Leaderboard	Ranking of users based on points and rank achieved	Competition, recognition
Level	Used to keep game space manageable and give a sense of progression. Completion rewarded with points and rank ups. Includes quizzes	Reward, status, competition, achievements
Progress bar	Used to track overall goal progression with the different modules	Feedback, achievement
Story/Narrative	Used to add meaning, provide context, and guide action. Comprises the elements of characters, plot, tension, and resolution	Foster learning, immersion, attitude change
Rewards Marketplace	Chat where users can redeem rewards in exchange of points	Reward, feedback, competition
Time restriction	Used to instill pressure	Sense of urgency
Social prod	Action of minimal effort to create a social interaction (e.g., "Like", "Deduct points")	Social
Activity feed	Stream of recent events in the course	Foster a sense of community and lead to feelings of recognition
Conformity anchor	The system informs the user how close a user is to the social norm (i.e., tutorial real-time performance)	Feedback, social
Animated feedback	Animated visual aid used to help users locate something in a system	Feedback
Verbal / visual / sound effects	Used to tell users about the state of the system as a result of interacting with the tutorial and modeling tool	Feedback
Avatar	Graphical representation of a user profile that users can customize to their preferences	Encourage self-expression, ownership
Game master	Observes and orchestrates the learning process. Assists or influences the behavior of learners	Feedback, social

The tutorial makes use of science fiction to present the educational material (*Story/Narrative*). In particular, the learner begins as a Rookie security requirements analyst hired by NovoX, an Intelligence Agency, to protect the Empire's interests across the Galaxy through designing a secure socio-technical system. By successfully completing the training material, the learner is awarded *Points*, which determine their *Rank* and their position in the *Leaderboard*. The learner can achieve up to 5 Ranks in their quest to become "Protector of the Galaxy", a position awarded to a few distinguished officials. The *Point System* awards points based on the estimated time the lessons and exercises take, whereas the *Rank* is determined using an algorithm that makes it harder to rank up as they obtain more points (and thus become more skilled). Note that the inner workings of the Points and Rank system are hidden to learners.

The tutorial also features the so-called *Rewards Marketplace*, where learners can exchange points in favor of 1) chatting with the *Game Master* and Security Requirements Experts or 2) *claiming* other *Rewards* such as: deduct points to another player, get personalized help in case of problems, purchase contextually-relevant hints, etc.

IV. PRELIMINARY RESULTS

We report preliminary findings from empirical studies in a non-experimental setting that focused on a subjective evaluation of the effectiveness of the STS-Tooltorial on learner *engagement*, *performance*, *intention to learn*, and *preferred learning environment*. The artifact was also evaluated in terms of overall *usability* and *satisfaction*.

We have conducted two separate empirical evaluations with (i) eight postgraduate information science students, and (ii) five IT professionals with three to 12 years of experience. The evaluation with students was conducted at Utrecht University in two separate sessions following the same procedure and setting. In particular, students were grouped in a room and requested to complete as much of the tutorial as possible within a 90-minute timeframe, under the supervision of the authors who were not to intervene unless, e.g., a participant redeemed a reward for hints. We hoped to induce (social) pressure by the combination of the time restriction, showing the leaderboard in real-time on a large screen, and having a Game Master physically present indirectly judging the modeling activities.

Since results of this evaluation suggested that physical co-location of the learners was not a key factor to have learners complete the tutorial, we decided to organize the evaluation with professionals in an online, remote fashion, with participants setting their pace independently.

In both evaluations data was gathered via pre-post surveys and through the analytics module of the interactive platform. In addition, we followed up after a week with some of the participants to brief them on their results and to gather more qualitative feedback on their responses and behavior.

Next, we examine the early findings across the different constructs of interest: engagement, usability and overall satisfaction, intention to learn, and preferred learning environment. To do so, we used the questionnaire available online[3]. No

[3]http://bit.ly/ststooltorialposttest

comprehensive statistics are reported here due to the inherent noise present in our small-sized sample.

Below, we use the following abbreviations taken from the APA guidelines for reporting statistical results: *M* for mean, *SD* for standard deviation, *r* for Pearson's correlation, *n* for sample size, *p* for the p-value, and *t* for the statistical t-test.

A. Engagement

The measure of engagement was mainly underpinned by the Short Flow Scale [34] ranging from 1 (minimum flow) to 7 (maximum flow). The student group self-reported a higher-than-average flow during the completion of the tutorial ($M = 4.69$, $SD = 1.03$), with no significant difference with respect to the professionals' flow ($M = 4.5$, $SD = 0.90$).

In both groups, the relationship between engagement and the number of points awarded ($r = .64$, $n = 13$, $p = .09$) and page views ($r = .65$, $n = 13$, $p = .08$) fell slightly short of statistical significance, given that the p value is above 0.05.

B. Usability and overall satisfaction

Usability was measured using the System Usability Scale (SUS): a valid, reliable instrument widely used in industry and academia. The aggregated SUS scores of both samples indicate that the interactive platform has *high usability* and is well integrated within the modeling tool ($M = 80$, $SD = 8.07$), with no significant difference among students and professionals; $t(11) = .16$, $p = .88$.

To measure satisfaction, half-way through the tutorial learners were asked to rate on an 11-item Likert scale how satisfied they were with the tutorial. Subsequently, in the post-evaluation survey, the Net Promoter Score (NPS) ranging from 0 to 10 was used as a proxy measure to satisfaction (as evidenced by intention to recommend the tutorial to a colleague). The groups reported high degrees of half-way satisfaction ($M = 7.38$, $SD = 1.19$) and intention to recommend ($M = 7.88$, $SD = 1.46$), with again no significant difference between professionals and students. In addition, both measures were found to be strongly correlated across groups ($r = .856$, $n = 13$, $p = .007$).

C. Intention to learn

We examined the extent to which the platform influenced the learner's intention towards learning more about SRE in general and the STS method in particular. Studying such intention is important for it has been shown to be a precursor of behavior [35]. We have used a 5-point Likert-type scale where we posed a statement saying that "I am interested in studying more about..." ranging from 1 (disagree) to 5 (agree).

Findings suggest that students had a significant higher intention to learn more about both SRE ($t(11) = 3.12$, $p = 0.01$) and the STS method ($t(11) = 2.15$, $p = 0.05$) than professionals did. In addition, a consistent pattern we observed across the groups was that learners had more intention to learn about SRE ($M = 3.15$, $SD = 1.14$) than the STS method in particular ($M = 2.77$, $SD = 1.01$), albeit the latter was the main focus of the tutorial; $t(11) = -2.13$, $p = 0.054$.

Overall, learners expressed *at least* an average-to-positive interest in pursuing further learning, with students being significantly more positive than IT professionals.

D. Preferred learning environment

Participants were finally asked to rank their preferred learning environment among four choices: massive open online course (MOOC), Book/Self-study, Built-in tutorial (e.g., our platform), and Classroom setting. We use a 4-point Likert-type scale ranging from 1 (most preferred) to 4 (least preferred).

While books were consistently ranked as the least preferred option by 62.5% of the participants ($M = 3.25$, $SD = 1.16$), the platform ($M = 2.13$, $SD = 1.13$) ranked first 37.5% of the time, closely followed by traditional classroom ($M = 2.25$, $SD = 1.04$) and MOOCs ($M = 2.38$, $SD = 1.06$). The results suggest that this innovative teaching method may be a competitor for both traditional and modern approaches.

V. Discussion and Future Directions

Overall, we find that the platform shows promising results across different key areas. All participants completed between 80% and 100% of the content with success, thereby reaching the intended learning outcomes within the time-frame that we set for them. Self-reported measures of usability, engagement, satisfaction, and intention to learn and recommendation were positive and robust for both students and professionals, which entails that the platform can cater to diverse audiences. However, improvements are certainly possible and necessary: our effort should be seen a first-of-a-kind attempt to use in-tool tutorials in (security) requirements engineering.

As for game elements, follow-up interviews highlighted the real-time interaction with the modeling tool, the storyline and the delivery format as key success factors. Extrinsic traditional elements such as points, badges, and leaderboards were not found to be highly interesting to participants, as the narrative and exercises (i.e., intrinsic factors) were engaging per se [36]. This is very much aligned with the recent trend on rethinking gamification from individual game elements to delivering a holistic, immersive game experience [14].

Interviews with professionals revealed that they found the method and modeling language too complex to use in a real context, where diagrams would tend to grow to the point of being unmanageable. This explains why we noticed a significant difference in the intention to learn about SRE and the STS method compared to students. We should investigate further whether this perception is shared by other practitioners to better understand if this was due to the language itself or to some features of our tutorial. It is worth mentioning that the second author delivers full-day trainings with STS-ml that rely on an interplay of traditional teaching and hands-on sessions, while the in-tool tutorial has a more limited time span.

These early findings, although preliminary and suggestive at best, indicate that incorporating a gamification-powered platform into the teaching process leads to good results and possibly better reception of the materials by the learners. Interestingly, we can conclude that this novel approach is perceived at least competitive with and as effective as other established teaching methods such as classroom settings, MOOCs or books/self-study for teaching conceptual modeling.

In the near future, we are planning to conduct a large-scale evaluation with graduate students, which we hope to report in the 2018 RE conference.

Our long-term roadmap includes covering the entire STS method with additional courses and extending the platform with other (S)RE-relevant modules. For example, we plan to use a similar tutorial for teaching about the recently released iStar 2.0 language for early goal-oriented RE [37]. An important research question to answer is whether certain modeling languages are better adequate for learning via interactive tutorials. In addition, since the STS-Tooltorial is currently coupled to the STS-Tool, the results reported here are tool-specific. The extent to which the underlying tool affects results is currently unknown and must be further investigated, both within and between modeling languages as well as the software tools that implement them.

We call for the joint support of the research community to help advance innovative teaching methods such as the presented one, in the hope of more effectively engaging and training the next generation of requirements engineers and conceptual modelers.

REFERENCES

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

[2] P. Devanbu and S. Stubblebine, "Software Engineering for Security: A Roadmap," *Proceedings of the Conference on the Future of Software Engineering (FOSE)*, pp. 227–239, 2000.

[3] W. Stallings and L. V. Brown, *Computer Security*. Prentice Hall, 2008.

[4] N. R. Mead, E. D. Hough, I. I. Stehney, and R. Theodore, "Security Quality Requirements Engineering (SQUARE) Methodology," DTIC Document, Tech. Rep., 2005.

[5] G. McGraw, *Software Security: Building Security in*. Addison Wesley Professional, 2006.

[6] E. Dubois and H. Mouratidis, "Guest Editorial: Security Requirements Engineering: Past, Present and Future," *Requirements Engineering*, vol. 15, no. 1, pp. 1–5, 2010.

[7] Y. Wand, D. E. Monarchi, J. Parsons, and C. C. Woo, "Theoretical Foundations for Conceptual Modelling in Information Systems Development," *Decision Support Systems*, vol. 15, no. 4, pp. 285–304, 1995.

[8] D. Basin, M. Clavel, J. Doser, and M. Egea, "Automated Analysis of Security-design Models," *Information and Software Technology*, vol. 51, no. 5, pp. 815–831, 2009.

[9] F. Dalpiaz, E. Paja, and P. Giorgini, *Security Requirements Engineering: Designing Secure Socio-Technical Systems*, 1st ed. MIT Press, 2016.

[10] J. B. Biggs and C. Tang, *Teaching for Quality Learning at University*. Open University Press, 2011.

[11] G. Sedrakyan, M. Snoeck, and S. Poelmans, "Assessing the Effectiveness of Feedback Enabled Simulation in Teaching Conceptual Modeling," *Computers & Education*, vol. 78, pp. 367–382, 2014.

[12] H. J. Nelson, G. Poels, M. Genero, and M. Piattini, "Quality in Conceptual Modeling: Five Examples of the State of the Art," in *Data and Knowledge Engineering*, vol. 55, no. 3, 2005, pp. 237–242.

[13] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, "Active Learning Increases Student Performance in Science, Engineering, and Mathematics," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8410–8415, 2014.

[14] S. Deterding, "Eudaimonic Design, or: Six Invitations to Rethink Gamification," in *Rethinking Gamification*, 2014, pp. 305–323.

[15] E. Paja, F. Dalpiaz, and P. Giorgini, "Modelling and Reasoning about Security Requirements in Socio-technical Systems," *Data & Knowledge Engineering*, vol. 98, pp. 123–143, 2015.

[16] A. Hoffmann and R. Weißbach, "ImProject: Improving RE Training by Combining it with Improvisation Theatre Sessions," in *Proceedings of the International Workshop on Requirements Engineering Education and Training (REET)*, 2014, pp. 11–19.

[17] D. Zowghi and S. Paryani, "Teaching Requirements Engineering through Role Playing: Lessons Learnt," in *Proceedings of the IEEE International Conference on Requirements Engineering (RE)*, 2003, pp. 233–241.

[18] P. Liang and O. De Graaf, "Experiences of Using Role Playing and Wiki in Requirements Engineering Course Projects," in *Proceedings of the International Workshop on Requirements Engineering Education and Training (REET)*, 2010, pp. 1–6.

[19] D. Damian and B. Al-ani, "Instructional Design and Assessment Strategies for Teaching Global Software Development: A Framework," in *Proceedings of the International Conference on Software Engineering (ICSE)*, 2006, pp. 685–690.

[20] L. Beus-Dukic, "Final Year Project: A Test Case for Requirements Engineering Skills," in *Proceedings of the International Workshop on Requirements Engineering Education and Training (REET)*, 2011, pp. 5–8.

[21] L. Mich, "Teaching Requirements Analysis: A Student Project Framework to Bridge the Gap between Business Analysis and Software Engineering," in *Proceedings of the International Workshop on Requirements Engineering Education and Training (REET)*, 2014, pp. 20–25.

[22] I. Park and M. J. Hannafin, "Empirically-Based Guidelines for the Design of Interactive Multimedia," *Educational Technology, Research and Development*, vol. 41, no. 3, pp. 63–85, 1993.

[23] H. Van der Meij, "Principles and Heuristics for Designing Minimalist Instruction," *Technical Communication*, vol. 42, no. 5, pp. 243–261, 1995.

[24] C. Plaisant and B. Shneiderman, "Show me! Guidelines for Producing Recorded Demonstrations," in *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2005, pp. 171–178.

[25] T. Grossman and G. Fitzmaurice, "ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding," *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI)*, pp. 1515–1524, 2010.

[26] Y.-K. Chou, *Actionable Gamification: Beyond Points, Badges, and Leaderboards*. Octalysis Media, 2016.

[27] S. O'Donovan, J. Gain, P. Marais, S. O. Donovan, and P. Marais, "A Case Study in the Gamification of a University-level Games Development Course," in *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (SAICSIT)*, 2013, pp. 242–251.

[28] B. De Schutter and V. V. Abeele, "Gradequest–Evaluating the Impact of Using Game Design Techniques in an Undergraduate Course," in *Proceedings of the International Conference on the Foundations of Digital Games (FDG)*, 2014.

[29] D. Dicheva, C. Dichev, G. Agre, and G. Angelova, "Gamification in Education: A Systematic Mapping Study," *Educational Technology and Society*, vol. 18, no. 3, pp. 75–88, 2015.

[30] C. C. I. Muntean, "Raising Engagement in e-Learning through Gamification," *Proceedings of the International Conference on Virtual Learning (ICVL)*, pp. 323–329, 2011.

[31] K. Kapp, *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. Pfeiffer, 2012.

[32] E. Paja, F. Dalpiaz, M. Poggianella, P. Roberti, and P. Giorgini, "STS-Tool: Socio-technical security requirements through social commitments," in *Proceedings of the IEEE International Requirements Engineering Conference (RE)*, 2012, pp. 331–332.

[33] K. Werbach and D. Hunter, *For the Win: How Game Thinking Can Revolutionize Your Business*. Wharton, 2012.

[34] S. Jackson, A. J. Martin, and R. C. Eklund, "Long and Short Measures of Flow: The Construct Validity of the FSS-2, DFS-2, and New Brief Counterparts," *Journal of Sport & Exercise Psychology*, vol. 30, no. 5, pp. 561–587, 2008.

[35] I. Ajzen, "The Theory of Planned Behavior," *Organizational Behavior and Human Decision Processes*, vol. 50, no. 2, pp. 179–211, 1991.

[36] R. Ryan and E. Deci, "Self-determination Theory and the Facilitation of Intrinsic Motivation," *American Psychologist*, vol. 55, no. 1, pp. 68–78, 2000.

[37] F. Dalpiaz, X. Franch, and J. Horkoff, "iStar 2.0 Language Guide," *CoRR abs/1605.07767*, 2016. [Online]. Available: http://arxiv.org/abs/1605.07767