

Defeating Dragons in the Classroom: Role-playing Games for Teaching Communication and Decision Making Skills in Software Engineering

Tom Beckmann
Hasso Plattner Institute
University of Potsdam
tom.beckmann@*

Eva Krebs
Hasso Plattner Institute
University of Potsdam
eva.krebs@*

Marcel Garus
Hasso Plattner Institute
University of Potsdam
marcel.garus@*

Stefan Ramson
Hasso Plattner Institute
University of Potsdam
stefan.ramson@*

Robert Hirschfeld
Hasso Plattner Institute
University of Potsdam
robert.hirschfeld@*

Abstract

Education in software engineering typically employs a variety of educational formats, including coding exercises, project work, case studies, or lectures. One important aspect of software engineering that appears inadequately addressed by these formats is communication—this includes communication with other programmers, clients, managers, or experts of other disciplines.

To address this lack, we prototyped the design of a tabletop role-playing game (TTRPG) that gives players the opportunity to engage in technical or other challenging conversations in a safe, simulated setting. As in a typical TTRPG, a game leader gives two to three players a description of scenarios, acts as the non-player characters, and lets players react, forming a loop of cooperative storytelling. In first playtesting sessions, we find that the game manages to put players in situations that are unfamiliar and sometimes uncomfortable for them but does so in a way that is educational and still feels playful. We discuss avenues to make use of this format in a classroom setting for a large number of groups, as it heavily relies on the improvisational skills and experience of the game leader.

1. Introduction

Role-playing is an often-used method in education (Rao & Stupans, 2012), entertainment, or even health care. At its core, it involves a person pretending to be another person, either acting as that person or describing their actions as that person. This form of disassociation from one's own person allows players to consider a scenario in a different light, for example, by building empathy for the character they themselves act as or that other people act as.

In entertainment, tabletop role-playing games (TTRPGs) are one of the most famous instances of role-playing. Typically, a group of players and one game leader are gathered around a table for the duration of the play session. The players and the game leader tend to have game-mechanical aids that support the scenario that is being played. For example, in the TTRPG "Dungeons and Dragons" (DnD), players have a character sheet that describes their character, including their strengths and weaknesses. A set of polyhedral dice is used throughout the game to determine success of actions in the scenario, which introduces a degree of uncertainty that prompts players to be spontaneous and flexible.

Even for role-playing games where mechanics play an important role, the core of the game experience remains cooperative storytelling. The game leader describes a scene and players describe how they act in response to that scene. Depending on the scene, these actions may involve dialogs exchanged between players or so-called non-player characters (NPCs) that are brought to life through the game leader. As the game should provide a safe space for the players, the play session should allow them to act in ways that they may otherwise be uncomfortable with, such as portraying an outgoing, boisterous character.

*hpi.uni-potsdam.de

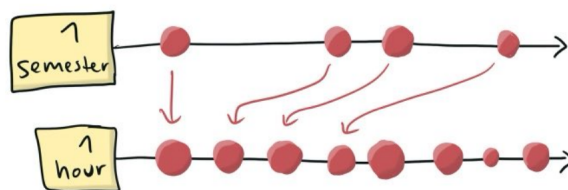


Figure 1 – Compared to a project that students work on over the course of a semester, similar interesting decision points can occur over the course of one hour in our game concept.

In this paper, we explore the players' ability to safely experiment with different roles through a prototype of a role-playing game for teaching software engineering. Players can, for example, assume senior or leadership roles in software engineering settings and are prompted to make decisions that the players may otherwise only get to make in a couple of years' time. Potentially, this may also help them understand and navigate scenarios where they are affected by such decisions in earlier stages of their career.

Communication plays an essential role in software engineering, as much of the engineering aspect encompasses activities that go beyond programming. To illustrate, a software engineering role may need to communicate with non-technical domain experts to understand requirements. Sources of confusion may include a lack of common vocabulary, as the domain experts use domain vocabulary and the software engineer uses software technical vocabulary. Or, goals may be misaligned, as the domain expert emphasizes aspects of the software that require an expensive technical investment that has not been properly accounted for during initial planning. Navigating these situations requires skills of careful communication, as well as an intuition on when to escalate or delegate points of discussion. While best practices and guidelines exist, the opportunities for putting these skills into practice are rare, especially in an educational setting, as illustrated in Figure 1. Our role-playing game prototype, consequently, offers players an opportunity to practice their communication skills and decision-making in a software engineering setting. In alignment with common game development practices, we conducted a first evaluation of our game prototype by performing playtests (Macklin & Sharp, 2016; Schell, 2019).

2. Role Playing in Education

Role-playing exercises are an established educational format (Rao & Stupans, 2012). Such exercises have been used to teach computer science topics that include but are not limited to: Requirements engineering, process management, quality assurance, software architecture, teamwork / soft skills, and software development (Hidalgo, Astudillo, & Castro, 2023). In order to make the most of a role-playing session, a reflection guided by a teacher afterwards is essential (Vykopál, Celeda, Svábenský, Hofbauer, & Horák, 2024). In the following, we will give an overview of how role-playing is used in computer science education, ranging from theater-like performances to scenario evaluations and traditional TTRPGs.

A common form of role-playing is acting as characters in a scene. This form of role-playing has, for instance, been used to teach agile development concepts (Bringula, Elon, Melosantos, & Tarrosa, 2019). Groups of three to four students received a prompt about a software team that uses agile development. The students then had to write their own script and act it out in ten to fifteen minutes. Students learned basic concepts of agile development and liked the cooperative nature of the exercise, but did not learn details about the topic. Additionally, shy students disliked the exercise and other students also noted that role-playing exercises are not an ideal tool for everyone, as they might, e.g., induce anxiety in students with special educational needs such as autism (Hidalgo et al., 2023).

Role-playing can also be used to simulate aspects of project work. For instance, one course aimed to increase communication between students in an online setting and teach about requirements engineering via role-playing (de Macedo, de Lima Fontão, & Gadelha, 2024). Each student team had two roles: they alternated acting as the development team working on the wishes of a customer team, and acting as that

customer team for other students. Students reported learning about teamwork and conflict resolution as well as online collaborative tools.

Commercial TTRPGs such as DnD have also been used in education. A course in a game development program included DnD exercises to enhance the students' communication skills (Veldthuis, Koning, & Stikkolorum, 2021). As it was part of a game-oriented program, the lecturers did not alter DnD, but instead embraced its nature as a game to also convey knowledge about storytelling and character creation to students.

In addition to exercises, role-playing has also been used to re-frame entire courses. One computer science course had each student create a character based on traditional TTRPG archetypes such as warriors, rogues, and wizards (Toth & Kayler, 2015). All class activities, including programming exercises, were altered to fit the theme. Students collected experience points from activities that were used for grading. Students liked that some of the new themed tasks, e.g., collecting special items on campus, encouraged them to talk to new students outside of established peer groups. Since the characters were separate from the students, including different names, the course could also add a leaderboard to encourage competition while also keeping students anonymous, if preferred.

Beyond role playing, another format with similar goals presented students with concrete requirements for a software system and they had to implement the interfaces of components by writing on physical index cards and connecting them via strings when they called each other (Blackwell & Arnold, 1997). When requirements change, students would have to untie or cut strings, helping to visualize the value of refactoring and avoiding coupling in system design.

Another alternative to role playing is storytelling—commonly, teaching experiences of how to address complex situations is done through a talk held by an experienced person. These can condense the interesting points and help to learn by example.

Undecided? is a board game that, similarly, highlights points of decision made by a team (Becker, Tsang, Booth, Zhang, & Fagerholm, 2020). By selecting decisions of a set of pre-determined choices, players cause effects that may surface in the short-term or only in the long-term.

3. Game Setup and Rules

Our game prototype aims to bring players into a position where they have to exercise planning, decision-making, and communication skills. We designed the rules for the game through iterative prototyping. We improvised a first session with one game leader and two players who defined a majority of the rules. In two subsequent sessions, we refined the rules and tried out slight variations to arrive at the ruleset we describe below. Notably, a number of problems with the ruleset as described below exist that we discuss in Section 5.

3.1. Setup

Most of the game's setup happens ad-hoc at the table with only a minor burden for preparation on the game leader.

Game Leader Preparation The game leader should prepare the scenario by deciding on or asking the players for themes that the group wants to explore as part of the play session. These themes may include activities such as hiring, crunch time, service outage response, or onboarding. Based on the themes, we recommend that the game leader prepare rough notes for potential complications the players might run into. Attempting to predict solutions to each complication proved to be a useful manner to derive possible further complications.

Role and Character Assignment When the group is gathered, the game leader assigns roles to each player. We recommend that these should imply distinct competencies and potentially also weaknesses, such that it is easier for players to act as their character. For example, if a character possesses extensive

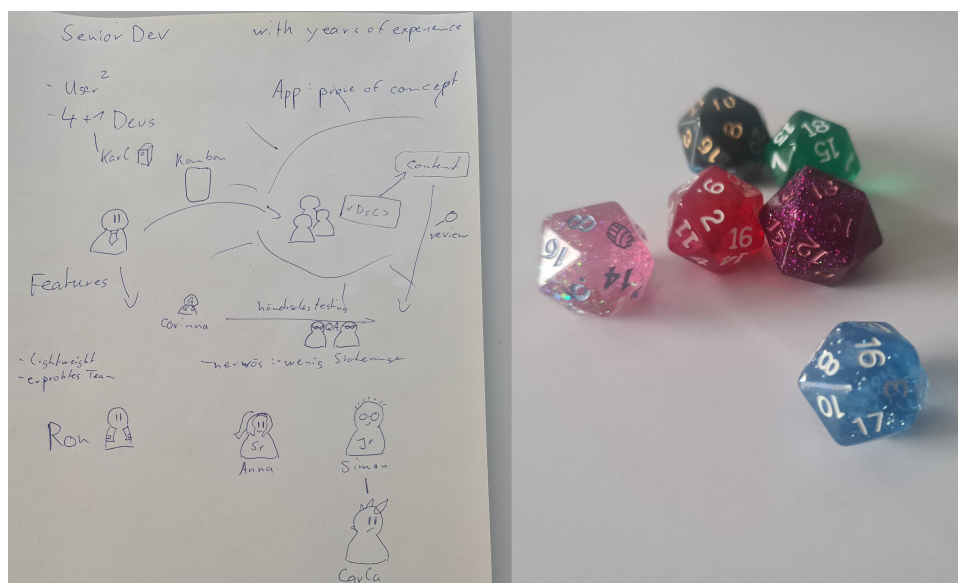


Figure 2 – Left: notes a player took during a play test, including names of characters, their relationships, and processes. Right: 20-sided dice, used in our prototype when outcomes are determined by chance.

experience with migrating cloud infrastructures, this enables the player to act as if they had multiple years of professional work experience in the relevant field without the player themselves knowing all required nuances. Assigning distinct competencies or roles among the players facilitates turn-taking, as players naturally hand the word to other players when their character's area of expertise would be helpful.

Scenario Description Once the roles are established, the game leader proceeds to describe the scenario the players are in: this may include the company they work at, colleagues they might refer to, and the team's current work mode. This description can already imply or state tension that exists in the team's setup, such as limited availability or suboptimal handovers between teams.

3.2. Play

With the characters and scenario established, the game loop begins.

Game Loop The structure of the game, following classic TTRPG rules, can be roughly summarized as follows, also illustrated in Table 1:

1. The game leader describes a setting, consequence, or action.
2. The players ask clarifying questions, if needed.
3. The players either react in-character, describe a reaction, or first discuss among themselves on how to best react, typically out-of-character.
4. The game leader considers the players' reaction and either repeats the loop or asks for a dice roll to determine its outcome before proceeding.

Dice Rolls In our prototype, we used a 20-sided die as seen in Figure 2 to add an element of uncertainty and chance to actions players take. This also supports the game leader, as outcomes to actions that are

Table 1 – Example Sequence of Play

Game Lead establishes scenes, describes "the world"	<i>"Your colleague from HR comes in and asks what to put into the new tech role's job description."</i>
Players describe or act out their actions	<i>"I tell him I'll send a text. I describe the project, tech stack, and I ask the colleague to add text about the company."</i>
Game Lead calls for dice rolls in important moments	<i>"To determine who applies, roll a die." (given the preparation, the Lead adds +5)</i>
Game Lead narrates results	<i>"It's a 5 (total 10). You get 50 applications." (likely good candidates, but challenging to decide who to interview)</i>

risky or uncertain do not have to be arbitrarily decided by the game leader. We encourage the game leader to consider the outcome of the dice roll as a tendency, rather than a scale from failure to success. For example, preparation that players took to mitigate risks with the decision should weigh more strongly than a 1 or 20 on the dice roll. Still, a 1, as the worst possible outcome of the dice roll, could present an interesting complication for the players but the game leader should still take care to realistically take into account the stakes of the decision and the players' preparation.

In addition, the game leader may use dice rolls to determine resources or outcomes outside of the players' control. These could include setting the size of a budget if it appears less relevant to the theme the game leader wants to bring across, or the number of applicants to a job posting the players created.

Scenes As described earlier, players and the game leader can either describe how characters act or act out interesting scenes. We encourage the game leader to offer the players opportunities to engage in acted dialog when the game leader thinks the conversation could include interesting challenges but not to force players to do so.

To reach interesting decisions throughout the game, the game leader will need to seek and often improvise adverse actions or circumstances to the players' plans. For example, a supplier may not be on time for an otherwise perfect plan. Or, the players' open communication with a customer may contradict promises another representative of their company has made without their knowledge.

4. Test Runs

We conducted four test runs, also known as playtests, of the game rules.

4.1. First Test Run

During the first test run, three of the co-authors played together, one being the game leader, and two players. The players assumed roles of a CTO at a startup and a Senior Developer who was just hired to bring in expertise for preparing the startup's infrastructure for a much larger scale.

Themes that were touched upon during play included defining roles and distributing or claiming responsibilities, hiring new team members, and mediating personal conflicts during onboarding. In total, the session took 45 minutes, with another 20 minutes afterward for reflection on the play session and the game system.

The players described feeling challenged or uncomfortable with decisions in multiple of scenes or conversations during the game but also that this was likely a good sign, as they appeared at the same time like they might happen in a real work setting. The game thus allowed them to explore the game world's reaction to their decisions without real-world consequences. When asked, the players listed a number of decisions they took throughout the game and indicated that the game encouraged reflection on these decisions.

4.2. Second Test Run

For the second test run, the same game leader led the game for three students in their last Bachelor's semester. This time, the players assumed the roles of three experienced software developers at an IT consultancy. Their current project involved writing drivers and control software for an expensive installation at a zoo. During the project, another lucrative project popped up and their management asked the team if they could create a proof-of-concept on a short timeline.

Themes included testing setups for hardware that is not on-site, communicating timelines with customers, and team organization when faced with the spontaneous new project.

The players appeared less confident than in the previous run. On multiple occasions, they were recalling best practices from software engineering lectures they had recently taken. The game leader made attempts to probe the understanding of the best practices by asking high-level aspects of their applicability and implementation in the scenario. Player feedback on the session and learning success was generally positive but players were struggling to state what they learned when prompted.

4.3. Third Test Run

For our third test run, a different co-author took the game leader role, and three students were selected as players. The players assumed roles of experienced software developers in a larger company that wanted to branch out and develop a new digital communication solution.

Themes included challenging technical requirements, pressure for marketing teams, and coordination or delegating work across departments, as they moved into a product lead role early in the game.

Feedback from the players was mostly concerned with game mechanics and structure of the storyline, which we will address in the next section.

4.4. Workshop Test Run

As part of the PPIG'25 workshop, we were able to let participants of the workshop try the game concept and include the resulting insights as a fourth test run. We asked them to form groups of three, select a game leader among them, and gave them a prepared scenario to run over the course of 45 minutes. In total, we had three groups, of which two played in person and one online.

Feedback was generally positive but it was noted that the involved players were all experienced programmers who had doubts that it would work as well with less experienced programmers. One group was initially struggling with the vague guidelines that our game rules provide but eventually got comfortable with the setting.

For this test run, we gave additional inputs to the players as a one-page website: a summarization of the rules, a scale to help game leaders judge the outcome of dice rolls, and a more explicit framework for accounting for preparation. The additional scale for preparation suggested applying adjustments to dice rolls, ranging from -10 for no preparation and a risky task to $+5$ for a well-prepared team or a well-known task. It was difficult to judge the effect this additional hint had but one team noted that they found it unclear how to use these adjustments. The same team also noted that it is not obvious when dice rolls should be called for.

Another point of difficulty concerned our prepared materials: we suggested separate roles of a "CTO" and "senior developer"—which appeared too similar, such that it was unclear how responsibilities should be split up. One team circumvented the tasks' complexity by simply stating that they would use "AI" for everything. The game leader in that team was unsure how to properly react to that, highlighting the dependency on improvisational skill or technical expertise with the technologies that players want to employ.

5. Discussion

Based on observations and player feedback, we derived a number of observations on the game rules and their use as a teaching tool.

5.1. Dependency on Game Leader

The most relevant issue that became apparent was a strong dependency on the game leader: they would have to be comfortable with improvising reactions of characters and sufficiently knowledgeable of software engineering processes to create realistic scenes for players to engage with.

To attempt to lessen the burden, we created a set of "events". These events comprise an inciting incident, potential consequences for the players, and a model approach for how the players may address the issue, such that the game leader could prepare potential setbacks as the players attempt to resolve the event. In practice, it was difficult to apply the events while the game was running. In the third session, players had started exploring implications of integration with a third-party platform, but the game leader felt unprepared to follow that thread and shifted to a prepared event. Feedback from the sessions suggested creating even more comprehensive scenarios that would allow the game leader to "look up" what should happen in response to a player's unexpected suggestion, and, if missing, at least find inspiration in similar situations. These events could also feature instructions for dice rolls, to serve as examples of what situations may be significant enough for rolling dice.

Similarly, it became clear that it was difficult to improvise interesting and relevant complications to the players' actions. Without those, the game ended up as fast-paced enumeration of scenes where typically one or at most two decisions had to be taken by players before it was resolved. One notable difference we drew from a comparison to TTRPGs like DnD was the lack of movement through space that often prompts the game leader to add "fluff"—details that set the mood of a scene or describe the setting. These often inconsequential details can help the game leader to gain valuable seconds to consider the next important event and how they want to present it to the players. In our game, as the narrative tended to jump from important scene to important scene without much "fluff", the game leader was constantly put in a position of improvising the next events without much buffer.

This dependency on the game leader's improvisational skill and software engineering experience is especially relevant for the game's use in an educational setting, as it makes it difficult to run a session with only few people as game leaders available. Further, the experience and learning outcomes will thus heavily depend on the game leader's familiarity with the game session's themes, whereas for use in a classroom, a degree of consistency would be desirable.

5.2. Impact of Dice Rolls

As described earlier in Section 3.2, dice rolls should only imply a tendency. Throughout the three game sessions, both the game leaders and players struggled with a suitable interpretation of dice rolls. As the range from 1 to 20 suggests a linear scale from worst possible to best possible outcome, numbers 1 through 5 were commonly perceived to yield potentially catastrophic outcomes. As rolling a 1 through 5 has a 25% chance of occurring, this led to some jarring and unrealistic developments in the story.

We attribute this expectation in part to prior experience with other TTRPGs, where dice rolls are typically combined with so-called modifiers that model the character's aptitude for the desired action. In, for example, DnD, it is not uncommon to add a +10 to rolls that characters are experts in, skewing the scale heavily toward success.

As we wanted to keep the ruleset light, we omitted any sort of modifiers or character attributes. Rather, the game leader is encouraged to consider the character's prior experiences as preparation for a roll and adjust outcomes accordingly. A future iteration of the game could consider adding a flat bonus to all roles to skew the perception toward success or adopt a different means of adding an element of chance to outcomes. An example may be a die with textual labels as opposed to numbers, or placing a labeled scale on the table during play that assigns, e.g., "minor setback" to the numbers 2 through 6 and thus helps adjust player expectations.

While it would also be possible to drop dice rolls completely, we kept them in our test runs for two reasons. First, as described before, they remove some pressure from the game leader to take decisions that "hurt" the players, shifting the blame to random chance. Second, players unanimously reported that

the dice added a feeling of excitement to the game. Notably, we did not try a test run without dice rolls, so the excitement and degree of uncertainty could potentially also be gained just from the unpredictable narration of the game leader.

5.3. Player Decisions

The core purpose of the game revolves around presenting players with interesting decisions and their communication. These decisions may concern who they want to talk to, how they address an issue, or what preparation they take. At several points, players noted that they were missing knowledge to articulate the steps they want to take. For example, in the first test run, the player acting as the senior developer tasked with preparing the infrastructure for scaling was initially struggling to play as that role, as it was unclear to them how to best proceed. In that instance, the game leader offered to present some general steps that they could consider and they proceeded to fill them out with how they figured it might make sense to proceed. Ultimately, while the game leader had to step in to help shape the decision, the player questioned and adapted the proposed steps, leading to interesting insights. This is in contrast to the second test run, where, in a similar situation, the players just said that they want to implement the steps offered by the game leader as suggested. It thus appears to be both a challenge for the game leader to volunteer information at the right level of abstraction and for the players to show willingness and interest to engage with the information. Still, it is clear that participants require a minimum level of experience to adequately derive actions. As one participant put it: the participants "don't know what [they] don't know".

As another concern, players and the game leader had a tendency to resolve conflicts quickly, which prevented interesting decisions from arising. Suggestions from the workshop to address this behavior included preparing "antagonists" in the story that may have conflicting interests with the players or giving players roles that initially appear to follow conflicting goals.

5.4. Other Limitations

In our test runs, we have only played the game with people who had already been working with one another for at least several months and trusted each other. In a classroom setting, it is unlikely that this will always be given. It may thus be necessary to add rules or mechanisms that protect players and ensure that they can act openly and experiment with decisions. Commonly, TTRPGs recommend players to conduct a so-called "Session Zero", where, among other things, themes and topics are collected that should not appear in the game's story. Further, a Session Zero often introduces safety tools that allow players to interrupt the game and discuss elements of the story that they felt uncomfortable with.

6. Conclusion

In this paper, we presented a concept and prototype for a game designed to let students experiment with communication and decisions in a software engineering context in the safe context of a game. Findings from three test runs indicate that the game is effective at engaging players with important decisions and placing them in situations that they would usually only encounter at later career stages. For actual use in a classroom setting, the most relevant unsolved challenge remains to support the game leader to improvise interesting consequences and scenarios, as well as provide them with sufficient knowledge of the software engineering themes they want to explore in the game.

7. References

- Becker, C., Tsang, T., Booth, R., Zhang, E., & Fagerholm, F. (2020). Undecided? a board game about intertemporal choices in software projects. In *Proceedings of the 31st annual meeting of the psychology of programming interest group* (pp. 122–133). Retrieved from <https://ppig.org/papers/2020-ppig-31st-becker/>
- Blackwell, A. F., & Arnold, H. L. (1997). Simulating a software project: The pop guns go to war. In *Proceedings of the 9th annual meeting of the psychology of programming interest group* (pp. 53–60). Retrieved from <https://www.cl.cam.ac.uk/~afb21/publications/PPIG97.html>

- Bringula, R., Elon, R., Melosantos, L., & Tarrosa, J. R. (2019). Teaching agile methodology through role-playing: What to expect and what to watch out. In *Proceedings of the 3rd international conference on education and multimedia technology* (p. 355–359). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3345120.3352733> doi: 10.1145/3345120.3352733
- de Macedo, G. T., de Lima Fontão, A., & Gadelha, B. (2024). Building soft skills through a role-play based approach for requirements engineering remote education. *J. Braz. Comput. Soc.*, 30(1), 1–16. Retrieved from <https://doi.org/10.5753/jbcs.2024.3071> doi: 10.5753/JBCS.2024.3071
- Hidalgo, M., Astudillo, H., & Castro, L. M. (2023). Challenges to use role playing in software engineering education: A rapid review. In H. Florez & M. León (Eds.), *Applied informatics - 6th international conference, ICAI 2023, guayaquil, ecuador, october 26-28, 2023, proceedings* (Vol. 1874, pp. 245–260). Springer. Retrieved from https://doi.org/10.1007/978-3-031-46813-1_17 doi: 10.1007/978-3-031-46813-1_17
- Macklin, C., & Sharp, J. (2016). *Games, design and play*. Boston, MA: Addison-Wesley Educational.
- Rao, D., & Stupans, I. (2012). Exploring the potential of role play in higher education: development of a typology and teacher guidelines. *Innovations in Education and Teaching International*, 49(4), 427–436. Retrieved from <https://doi.org/10.1080/14703297.2012.728879> doi: 10.1080/14703297.2012.728879
- Schell, J. (2019). *The art of game design: A book of lenses* (3rd ed.). London, England: Routledge.
- Toth, D., & Kayler, M. (2015). Integrating role-playing games into computer science courses as a pedagogical tool. In A. Decker, K. Eiselt, C. Alphonse, & J. L. Tims (Eds.), *Proceedings of the 46th ACM technical symposium on computer science education, SIGCSE 2015, kansas city, mo, usa, march 4-7, 2015* (pp. 386–391). ACM. Retrieved from <https://doi.org/10.1145/2676723.2677236> doi: 10.1145/2676723.2677236
- Veldhuis, M., Koning, M., & Stikkorum, D. R. (2021). A quest to engage computer science students: Using dungeons & dragons for developing soft skills. In *CSERC '21: The 10th computer science education research conference, virtual event, the netherlands, november 22 - 23, 2021* (pp. 5–13). ACM. Retrieved from <https://doi.org/10.1145/3507923.3507927> doi: 10.1145/3507923.3507927
- Vykopal, J., Celeda, P., Svábenský, V., Hofbauer, M., & Horák, M. (2024). Research and practice of delivering tabletop exercises. In M. Monga, V. Lonati, E. Barendsen, J. Sheard, & J. H. Paterson (Eds.), *Proceedings of the 2024 on innovation and technology in computer science education v. 1, iticse 2024, milan, italy, july 8-10, 2024*. ACM. Retrieved from <https://doi.org/10.1145/3649217.3653642> doi: 10.1145/3649217.3653642