Gamifying and Conveying Software Engineering Concepts for Secondary Education: An Edutainment Approach

Roy Oberhauser
Computer Science Dept.
Aalen University
Aalen, Germany
roy.oberhauser@hs-aalen.de

Abstract—**Because of its abstract nature, software engineering faces image, perception, and awareness challenges that affect its ability to attract sufficient secondary school age students to its higher education programs. This paper presents an edutainment approach called Software Engineering for Secondary Education (SWE4SE), which combines short informational videos and a variety of simple digital games to convey certain SE concepts in an entertaining way. Our realization mapped various SE concepts to seven digital games, and results from an evaluation with secondary students showed that a significant improvement in the perception, attractiveness, and understanding of SE can be achieved within just an hour of play. Thus, we suggest that such an edutainment approach is economical, effective, and efficient for reaching and presenting SE within secondary school classrooms.**

Keywords-software engineering education; software engineering games; game-based learning; digital games.

I. INTRODUCTION

The demand for software engineers appears insatiable, and computer science (CS) faculties and the software engineering (SE) discipline appear to be steadily challenged in attracting and supplying sufficient students to fulfill the demand. While it may appear to each higher education institution and country to be a local or regional problem, the challenge may indeed be more common and broader in nature. For example, in the United States in 2005, after a boom beginning around 2000, a 60% decrease over 4 years in the number of freshmen specifying CS as a major was observed [1]. And US bachelor degrees in CS in 2011 were roughly equivalent to that seen in 1986 both in total number (~42,000) and as a percentage of 23 year olds (~1%) [2]. As another example, Germany also observed a negative trend from a record number of CS students in 2000, and one 2009 news article [3] blamed the negative image of CS among the young generation. While the number of starting CS undergrads in Germany has since increased, roughly 33,000 software developer positions remain unfilled in 2014 [4]. In addition, the forecast demographic effects in certain industrial countries imply a smaller younger population available to attract, reducing the overall supply and thus increasing the competition between disciplines to attract students and workers. It is thus a critical and continual worldwide issue to attract young women and men to SE.

Concerning SE's image, according to D. Parnas [5] there is apparently confusion in higher education institutions as to the difference between CS and SE, and we assert this affects secondary education as well. The CS equivalent term in Germany, *informatics*, is much more publically known and marketed as a discipline or field of study than is SE. Thus, SE programs struggle in the overall competition between disciplines to attract secondary students for this critical area to society, since SE must first raise awareness about its field.

The concepts inherent in SE, as exemplified in the SWEBOK [6], tend themselves to be abstract and to deal with abstractions. Thus, they are difficult to convey, especially to secondary school students who have not significantly dealt with such abstractions, and cannot thus practically imagine what they mean. Furthermore, secondary school teachers and institutions are challenged in teaching CS, and have likely themselves not been introduced to SE.

Learning is a fundamental motivation for all game-playing, as game designer C. Crawford [7] once stated. With this in mind, this paper contributes an SE edutainment approach we call SWE4SE for gamifying and conveying SE concepts with secondary school students as the target audience. It describes the principles in the solution concept and example mappings of SWEBOK concepts onto relatively simple digital games, demonstrating its viability, and the evaluation with secondary students showed that combining short informational videos and a variety of simple digital games can be economical, effective, and efficient for improving SE awareness, perception, and attractiveness.

The paper is structured as follows: Section II describes related work. Section III describes the SWE4SE solution principles, our game design, and the incorporated SE concepts. Section IV describes our realization. Section V details the evaluation, followed by a conclusion.

II. RELATED WORK

Serious games [8] have an explicit educational focus and tend to simulate real-world situations with intended audiences beyond secondary education. [9] performed a literature search of games-based learning in SE and "found a significant dearth of empirical research to support this approach." They examine issues in teaching the abstract and complex domain of requirements collection and analysis and, more generally, SE. Using a constructivism paradigm, the role-playing client-server-based SDSim game has a team manage and deliver multiple SE projects. The systematic survey on SE games by [10] analyzed 36 papers, all of which had targeted undergraduate or graduates. Regarding secondary education, whereas initiatives for teaching programming are more common, conveying SE concepts in general and gamifying SE has not hitherto been extensively studied, nor has the educational value of explicitly "non-serious" (or fun) games for this population stratum.

Concerning the perception and attractiveness of CS among secondary students, the study by [11] of 836 High School students from 9 schools in 2 US states concluded that the vast majority of High School students had no idea what CS is or what CS majors learn. This conclusion can most likely be transposed to the lesser known discipline of SE.

In contrast, SWE4SE is targeted not towards higher education, but rather secondary school students with an explicit non-serious game approach. Our results compare with [11], but go further in showing that an edutainment approach can improve the perception and attractiveness of SE. Compared to other learning game approaches, it explicitly makes the tradeoff to value entertainment more and education less in order to retain student engagement and enjoyment. It also explicitly includes short informational and entertaining video sequences to enhance the experience beyond gaming alone.

III. Solution

SWE4SE consists of a hybrid mix of short informational and entertaining videos and a variety of relatively simple digital games. Our solution is necessarily based on certain assumptions and constraints. For instance, we assumed that the players may not only be playing in a compulsory classroom setting, but may play voluntarily on their own time, meaning that they could choose to stop playing if it becomes boring or frustrating and discard the game for some more interesting game. Thus, the edutainment is considered to be "competing" with available pure entertainment options. However, we expect that the game may be promoted to secondary school teachers where they would introduce students to the game, meaning that our concept must not necessarily compete solely with commercial products and other entertainment. We also assumed that the motivational factors for students in the SWE4SE are curiosity, exploration, discovering different games, and finding fun areas.

A. Solution Design Principles

Web-browser Principle (P:Web): To broadly reach the target audience (secondary students ages 12-18), we chose to focus our initial design and implementation on a web-based game solution and avoid the installation of components on game or various OS platforms. This constrains the available game options, but increases the reachable population.

Engagement / Enjoyment Principle (P:En): We want to keep the students engaged and to emotionally enjoy the experience. To reduce the likelihood of a player discontinuing due to lack of fun, we chose to value and prioritize the fun aspect more than pushing the learning of SE educational concepts. We are thus aware of the fact that less of the SE material may be conveyed and retained, but by retaining engagement over a longer timeframe, further possibilities for SE concept conveyance result.

Game Reuse Principle (P:GR): Leverage known games and game concepts (repurposing) when possible, such as those in [12]. Players may thus already know the basics of how the original game works - reducing the time to become proficient, and they may find the new twist involving SE concepts interesting. Also, more time and cognitive capacity may be available for the mapping of SE concepts to the game elements when compared with completely unfamiliar games.

Simple Game Principle (P:SG): Utilize relatively simple games when not utilizing already known games (cp. *P:GR*). This reduces the overall effort required to acquire game proficiency and to acquire the SE concepts.

SE Concept Reinforcement via Game Action Principle (P:GA): during the games, immediate feedback messages that reinforce game associations to SE concepts are given, e.g., "Correct, the quality was OK" or "Oops, the component was released with quality defects" for a software quality assurance (SQA) game. This makes it more transparent how choices and actions are reflected in SE concepts.

B. Edutainment Elements and SE Concept Mappings

We believe that certain aspects of SE cannot be conveyed well solely with games and should thus be supplemented.

Text components: a brief amount of onscreen text was used to introduce the topic area, relevant SE concepts, and the game objective and major game elements. Such a short text that can be clicked away does not overly interfere with the experience, and can be read or skimmed rather quickly. Using these, later bonus-level text questions can reference some prior text or video as a way to verify comprehension.

Video components: a short 5-minute informational video described how prevalent code is, society's dependence on software, and how important software development and software engineers are. The ability to include relevant videos, and the ability for users to explore and discover such videos, adds to the "adventure".

Game components: Various concepts from SWEBOK were chosen, with the selection subjectively constrained by our project resources, technical and game engine constraints, and how well a concept might map to a game concept. The selection, mapping, and prioritization of what to realize was subjectively reflected and decided on as a team, which is summarized in Table I.

TABLE I. SE Concept to Game mapping

SE Concept	Analogous Common Game	SWE4SE Game Variant
1) Processes	Pac-Man	ProcMan
2) Quality assurance	Pinball	Q-Check
3) Requirements	Tower Defense	ReqAbdeck
4) Testing	Angry birds	Angry Nerds
5) Construction	Angry birds	Reverse Angry Nerds
6) Defect remediation	Space invaders	Bug Invaders
7) Project management	Maze	Path Management

The mapping should be interpreted as exploratory and not prescriptive; our intention here is rather to demonstrate the possibilities available in such an edutainment approach.

IV. REALIZATION

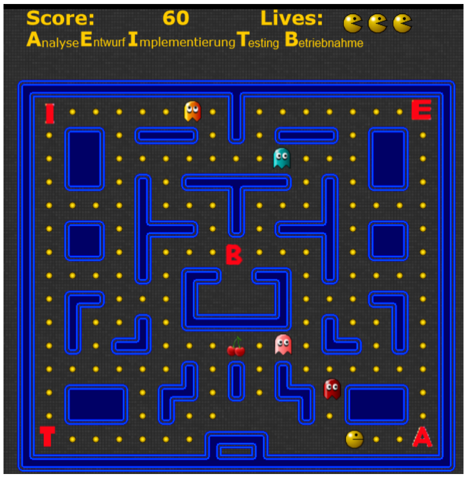
Scirra Construct 2 was used to develop the games. Layouts and Event sheets were used, and each game object type is given properties with certain states and behavior. Sounds and music were integrated. The web browser requires HTML5 and Javascript support. Text components were initially German, but could be readily internationalized. For brevity, details on how points are earned, bonus levels, speed changes, or lives lost in each game are omitted.

A. Conveying SE Concepts in the Various Games

For each SE concept below, how the analogous common game was mapped to corresponding primary game goals is described and depicted.

1) SE Processes: to convey an engineering process, we chose to introduce sequential activities common to many engineering processes. Based on the waterfall process, these were Analysis, Design, Implementation, Testing, and Operations (ADITO, or equivalently AEITB in German). We also provided a test-driven development (TDD) variant where the Testing occurs before Implementation (ADTIO).

ProcMan: this game is analogous to the well-known Pac-Man game (see Figure 1), with a twist that, whereas in PacMan one got points by traveling everywhere in the maze in any order, the goal here for the player is to follow a given SE process sequence by making ProcMan consume the distributed letters in the given order while avoiding ghosts.

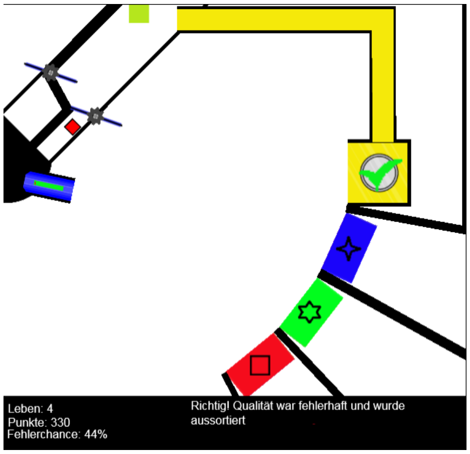

Score: 60 Lives:

AnalyseEntwurfImplementierungTesting Betriebnahme

Figure 1. ProcMan game conveys SE processes (screenshot).

2) Software quality assurance: SQA differs on the type of software component being inspected (e.g., GUI, database, business logic). Quality awareness and attention to detail matter, and the appropriate QA tools and testing procedures must be chosen.

Q-Check: this game is loosely analogous to pinball (see Figure 2). Software components (SoCos) portrayed as colored shapes spin and drop into a funnel, while a cannon (blue on the left) automatically shoots them individually after a certain time transpires (indicated via a decreasing green bar on the cannon). The player's goal is to select the process (tunnel on the right) that matches the SoCo type currently in the cannon based on both color and shape, or reject it for rework (yellow) if it is defective, improving the future error rate.

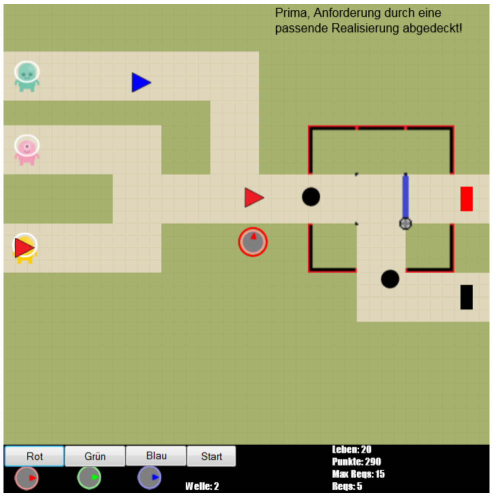

Leben: 4
Punkte: 330
Fehlerchance: 44%

Richtig! Qualität war fehlerhaft und wurde aussortiert

Figure 2. Q-Check game conveys SE quality assurance (screenshot).

3) Software requirements: this concerns itself with the SE concept of requirements coverage, for instance not overlooking a requirement, determining which requirements to fulfill how and when (different requirement types need different specialized competencies), which requirements to jettison (e.g., due to insufficient business value).

ReqAbdeck: (*Abdeckung* in German means "coverage") this game is analogous to the popular game Tower Defense (see Figure 3), whereby here waves of "reqs" (requirements as colored triangles) flow from left to right, and towers in various colors that cover (fire) only at their requirement color must be dragged to the appropriate vicinity before the "reqs" reach the gate. The towers disappear after a short time indicated on their border. Thus, one is not covering critical requirements in time with the matching implementation, ignoring or forgetting a requirement, or not dropping via a gate those requirements without business value (denoted by black circles). One example action message here is "Great, requirement was covered by a suitable realization.


Prima, Anforderung durch eine passende Realisierung abgedeckt!

Welle: 2

Leben: 20
Punkte: 290
Max Reqs: 15
Reqs: 5

Figure 3. ReqAbdeck conveys SE requirement coverage (screenshot).

ISBN: 978-1-61208-438-1

4) Software testing: the focus here is determining where to test to find deficiencies in some software construct.

Angry Nerds: this game is loosely analogous to the popular game Angry Birds (see Figure 4). For various reasons we chose to depict hardware-like testing here of children's blocks, since it was not obvious to us how to convey code-based testing in an obvious manner without necessitating extra explanations. The player's goal in this case is to test a given construct of slabs surrounding PCs by determining where and how hard to throw a mouse at it to knock it completely over. They realize that multiple tests are necessary to discover its weaknesses.

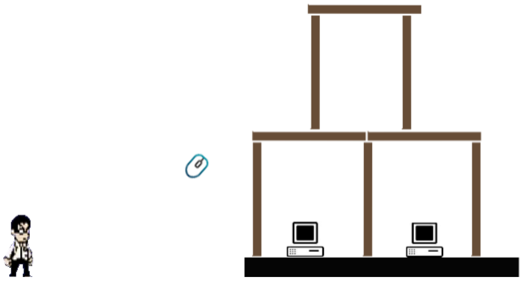


Figure 4. AngryNerds conveys SE testing (screenshot).

5) Software construction: the point here is to engineer or build the software such that it exhibits resiliency.

Reverse Angry Nerds: this is a bonus level of the previous game, and reverses the role as shown in see Figure 5, having the player now try to build a resilient construct by dragging and placing slabs in such a way that it withstands the automated testing (cannonball shot at it).

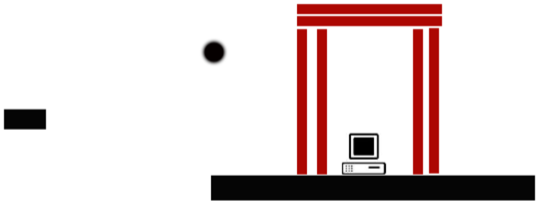


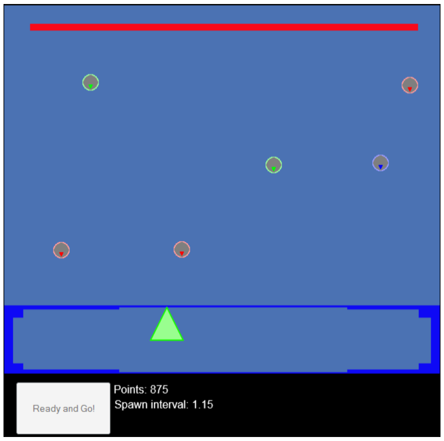
Figure 5. Reverse AngryNerds game conveys SE construction (screenshot).

6) Software defect remediation: the learning focus is that different defect types require different remediation techniques and countermeasures applied accurately.

Bug Invaders: in this game, analogous to space invaders, a matching remediation technique (ammunition color in the lower shooter) and firing accuracy are needed to destroy exactly that specific bug type that drops down before it creates project damage (see Figure 6).

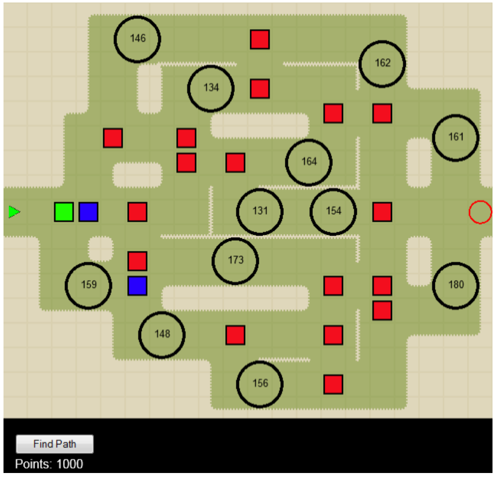
7) Software project management: here we convey that multiple choices towards optimizing project costs exist. With appropriate planning, the project goal can be reached with the allotted resources, while unexpected problems can be overcome but cost unplanned resources.

Path Management: in this game a player must manage a starting budget in points efficiently (see Figure 7). From the project start (green triangle) a path selection is made to take it to the end (red circle). Red blocks depict possible steps, blue steps the currently available choices, and green the current position. Each step costs 100 points, while randomly generated problems (black circles) add to the planned costs.



Points: 875
Spawn interval: 1.15

Figure 6. Bug Invaders convey SE defect remediation (screenshot).



Points: 1000

Figure 7. Path Management conveys SE project management (screenshot).

B. Realization of SE Exploration Concept

To tie it all together, the realization includes a SE universe to navigate to and discover various SE planets. Figure 8 shows the spaceship navigating in two dimensions. A shield level, reduced when colliding with asteroids, is shown as a colored line next to the ship. The game ends when the shields are lost or on collision with the sun. The bottom right of the screen shows a navigation map with the location of all planets (red first, green when visited, and violet for the home planet, and the spaceship as an arrow.

When arriving at a planet (Figure 9), a short text about SE concepts that relates to the game is shown, which when understood, can later be used to answer bonus questions at a gate. The portal to the game is shown on the left. The brown gate and fence shows a darkened advanced level area only accessible by successfully passing a gate requiring that SE challenge questions be answered correctly. This then enables passage and undarkens the upper bonus region top.

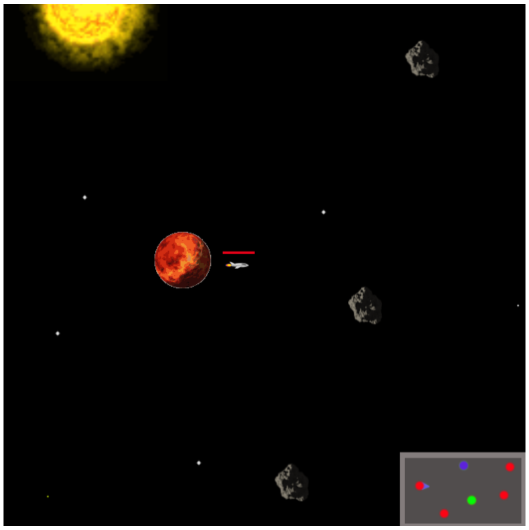


Figure 8. Spaceship navigating the SE universe (screenshot).

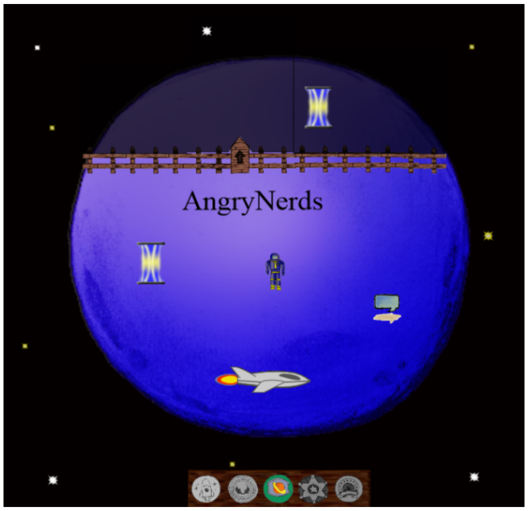


Figure 9. Example of a uniquely named SE game planet (screenshot).

On the home planet, a TV tower shows the video.

The realization is economical in that it can be widely distributed (*P:Web*) without client installation costs or large cloud server investments (it runs locally in the browser).

V. Evaluation

The convenience sampling technique [13], common in educational settings, was selected to evaluate our SE edutainment approach due to various constraints otherwise inhibiting direct access to a larger random population sample of the target population stratum. These constraints include logistical and marketing effort and budget, privacy issues, and acquiring parental consent for school age children.

Setting: Two teachers gave us access to 20 students in informatics interest groups for 90 minutes at two different public university preparatory (secondary) schools in the local region. *Setting A* using an alpha version of the software tested with a group of 8 males, and a later *setting B* using a beta version in a different city with 6 females and 6 males students. Figure 10 shows the age and gender distribution, and Figure 11 indicates their current game usage frequency.

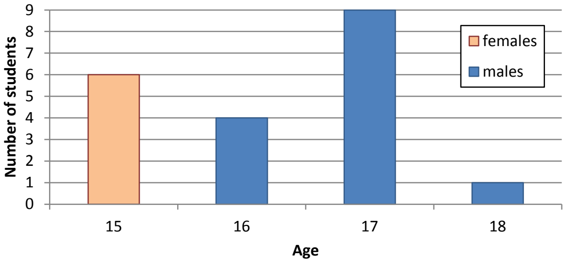


Figure 10. Student age and gender distribution.

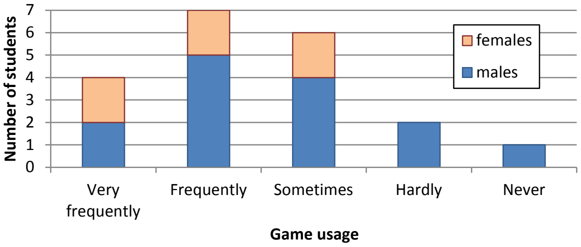


Figure 11. Prior game usage frequency distribution.

Questionnaire: While we considered utilizing the GEQ [14], it appeared more appropriate for more immersive games. Hence, due to the player ages, the limited time they had for playing multiple different short games (7 games in one hour), and the limited time, attention, and incentives for filling out pre- and post-questionnaires (10 minutes respectively), only a few questions about their state before and after with regard to negative or positive affect were included. They were asked but not compelled to answer all questions, so some fields were left blank by some students, which could be interpreted to mean they did not understand the question, did not know how to answer, or did not care to answer the question. Blank answers were thus omitted.

Session: The empirical evaluation consisted of 90-minute sessions held in two different settings A and B. The first 5 minutes consisted of a short introduction as to the purpose of our visit and what to expect, involving no teaching. Students were then given 10 minutes to fill out anonymous printed questionnaires in German that provided us with initial basic data. When all were done, they began their one-hour edutainment experience. In the 10 minutes directly thereafter, monitors were turned off and they completed the second part of their questionnaire, which focused on their experience and understanding, after which we held a 5-minute group feedback session.

Results: We observed that all students were engaged with the software for the entire hour and appeared to enjoy the experience *(P:En),* and occasionally interacted excitedly with fellow students. Below is our analysis of the questionnaire results. Unless otherwise indicated, averages were based on a scale of 1 to 5 (1 being very good, 5 bad):

- Overall experience: 2.1 (good); relates to *P:En*
- Game enjoyment: 2.0 (good); relates to *P:En*
- Helpful conveying several SE concepts via different games: yes (17), no (1); relates to *P:SG* and *P:GR*

ISBN: 978-1-61208-438-1

- Success rate in correctly recalling the SE concepts associated with each named game (open answers): 62%; relates to *P:GA* and Text components. Note that the game names in the questions could serve as a hint, but these did not include the complete and explicit SE concept nor was the game accessible.

Video:

- Watched the video attentively: yes (20)
- Video and its quality (good/bad): good (20)
- Video length 5 minutes: keep (19), lengthen (1)

Table II shows the change in perception, attractiveness, and understanding of SE after the experience.

TABLE II. CHANGE IN SE PERCEPTIONS

Change in responses	Before	After	Improvement
Importance of SE for society[a]	1.7	1.2	33%
Attractiveness of SE as a personal career path[b]	3.3	2.7	16%
Ability to define what SE is[c]	2.9	2.3	20%

a. Scale of 1 to 3 (1=very important, 3=not important); 2 wrote "don't know" in the prequestionnaire.
b. Scale of 1 to 5 (1=very attractive, 5=not attractive)
c. Answer graded (1 excellent, 2 very good, 3 satisfactory, 4 sufficient) for B group only.

As to interpreting the results, a convenience sample can obviously contain a number of biases, including under- or overrepresentation. Our supervision of the evaluation process and physically observing that the software was actually used for an hour by each of the students separately, and that each questionnaire was individually filled out, removed certain other kinds of threats to validity.

The evaluation showed the effectiveness of the approach: because students in this age group had previous familiarity with gaming, they had no difficulty rapidly discovering and playing the games intuitively without training or help, they understood the intended mapping of SE concepts onto game elements, and the perception, attractiveness, and understanding of SE improved significantly without discernable gender differences. It was efficient in achieving these effects in the relatively short span of an hour of usage.

In summary, an edutainment approach with short videos, short text components, and a variety of simple games appears promising for effectively and efficiently improving the awareness about and image SE, at least for our target population stratum.

VI. CONCLUSION AND FUTURE WORK

We described SWE4SE, an edutainment approach for gamifying and conveying software engineering concepts to secondary school students. Various principles used of the edutainment approach were elucidated, and it was shown how various SE concepts could be mapped and realized with various digital game concepts and elements. The evaluation showed that an edutainment approach, combining short videos and text elements, and a variety of simple digital games, can be promising for improving SE awareness in our target population stratum. Since this target age group is already familiar with gaming and utilizes gaming relatively frequently, the approach appears reasonable for reaching a larger populace.

A challenge remains in making secondary students aware of the availability the edutainment and motivating them to utilize it on a direct or individual basis. While social networks appear feasible for raising awareness, in the face of the abundance of entertainment and game options available, we believe that the most promising approach will likely be informational publicity campaigns towards informatics teachers in secondary schools where groups (i.e., interest groups or classrooms) utilize the software together in a structured setting.

Future work will include public access on the university website, enabling integrated data collection and web analytics to provide further insights into how users became aware of the edutainment, which games were utilized for how long, the number of return visitors, and the inclusion of online surveys. Additionally, the SE pool of questions will be expanded and question and answer placement randomized. A point and badge ranking of top players may provide a separate incentive and motivation for certain player types.

ACKNOWLEDGMENT

The author thanks Carsten Lecon, Christian Wilhelm, Flamur Kastrati, and Lisa Philipp for their assistance with the concepts, realization, and graphics.

REFERENCES

[1] J. Vegso, "Interest in CS as a Major Drops Among Incoming Freshmen," Computing Research News, vol. 17, no.3, 2005.

[2] B. Schmidt, http://benschmidt.org/Degrees/ 2015.07.04

[3] U. Hanselmann, "Die große Kraft," Engl: "The major force", Die Zeit, No. 22, 2009. http://www.zeit.de/2009/22/C-Faecherportraet-Informatik/komplettansicht 2015.07.04,

[4] Bitkom, "In Deutschland fehlen 41.000 IT-Experten," 2014. https://www.bitkom.org/Presse/Presseinformation/Pressemitteilung_1704.html 2015.07.10

[5] D. Parnas, "Software engineering programs are not computer science programs," Software, IEEE, 16(6), pp. 19-30, 1999.

[6] P. Bourque and R. Fairley, "Guide to the Software Engineering Body of Knowledge (SWEBOK (R)): Version 3.0," IEEE Computer Society Press, 2014..

[7] C. Crawford, "The art of computer game design," McGraw-Hill/Osborne Media, 1984.

[8] C. Abt, "Serious Games," The Viking Press, 1970.

[9] T. Connolly, M. Stansfield, and T. Hainey, "An application of games-based learning within software engineering," British Journal of Educational Technology, 38(3), pp. 416-428, 2007.

[10] C. Caulfield, J. Xia, D. Veal, and S. Maj, "A systematic survey of games used for software engineering education," Modern Applied Science, 5(6), 28-43, 2011.

[11] L. Carter, "Why students with an apparent aptitude for computer science don't choose to major in computer science," SIGCSE Bulletin, ACM, vol. 38, no. 1, Mar. 2006, pp. 27-31.

[12] S. Kent. "The Ultimate History of Video Games," Three Rivers Press, 2001.

[13] L. Given (Ed.), "The Sage encyclopedia of qualitative research methods," Sage Publications, 2008.

[14] W. IJsselsteijn, K. Poels, and Y. De Kort, "The Game Experience Questionnaire: Development of a self-report measure to assess player experiences of digital games," TU Eindhoven, Eindhoven, The Netherlands, 2008.