An Ontological Perspective on the Digital Gamification of Software Engineering Concepts

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

***Abstract* - Software engineering (SE), because of its abstract nature, faces awareness, perception, and public image challenges that affect its ability to attract sufficient secondary school (high 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 without necessitating or intending any skill development such as programming. Our realization mapped various SE concepts to seven digital games and these are described using the Software Engineering Body of Knowledge (SWEBOK) and game elements ontologies. We also investigated the maturity of ontologies for concepts in this area to determine the feasibility for a methodological mapping of SE concepts to game element and game logic to further automation. Results of SWE4SE with secondary students showed a significant improvement in the perception, attractiveness, and understanding of SE can be achieved within just an hour of play, suggesting such an edutainment approach is economical, effective, and efficient for reaching and presenting SE within secondary school classrooms. Our ontological investigation showed significant gaps towards formalizing or automating the approach.**

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

I. INTRODUCTION

This paper is an extended version of a paper presented at the Tenth International Conference on Software Engineering Advances (ICSEA 2015) [1].

As digitalization sweeps through society, the demand for software engineers appears insatiable. As attractive as a software development career may appear in the media or press, computer science (CS) faculties and the software engineering (SE) discipline in particular appear to be steadily challenged in attracting and supplying sufficient students to fulfill the job market demand.

To each higher education institution and country, it may appear to be only a local or regional problem, yet the challenge may indeed be more common and broader in nature. Statistics for SE are not as readily available as those for CS. Since SE can be viewed as being a subset of CS, and not all CS faculties offer a separate SE degree or concentration (but may include applicable SE courses and electives in their curriculum), we focus on available statistics for CS majors and assume they correlate to SE in this paper. 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 [2]. 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%) [3]. As another example, Germany also observed a negative trend from a record number of CS students in 2000, and one 2009 news article [4] 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 in Germany remain unfilled in 2014 [5]. In addition, the demographic effects forecast for various industrial countries imply a decreasing younger population, reducing the overall supply and thus increasing the competition between disciplines to attract students and eventually qualified workers. Thus, it remains a worldwide issue to attract young women and men to study SE.

Concerning SE's image, according to D. Parnas [6] 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 are exemplified in the SWEBOK [7]. The objectives of the SWEBOK include promoting a consistent view of SE, clarifying the bounds of and characterizing the contents of SE with respect to other disciplines, and providing a topical access to the SE Body of Knowledge. Knowledge areas (KAs) have a common structure, including a subclassification of topics, topic descriptions with references. The KAs with the abbreviations used in this paper are shown in Table I and Related Disciplines and shown in Table II. These KAs 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.

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Learning is a fundamental motivation for all game-playing, as game designer C. Crawford [8] once stated. Computer games involving the learning of some knowledge area are called educational computer games [9]. Both persuasive games [10] and serious games [11][12] have been used with the intent to change behavior or attitudes in various areas. With respect to engagement and learning, a study by [13] identifying 7392 papers that included 129 with empirical evidence found that the most frequently reported outcome and impact of playing games were affective and motivational as well as knowledge acquisition or content understanding. Thus, a gamification and edutainment fun approach targeting secondary school students appears promising for addressing both the aforementioned SE attractiveness and learning about SE concepts and the SE discipline in general.

TABLE I. SWEBOK KNOWLEDGE AREAS (KA)

Knowledge Area	Abbreviation
Software requirements	SR
Software design	SD
Software construction	SC
Software testing	ST
Software maintenance	SM
Software configuration management	CM
Software engineering management	EM
Software engineering process	EP
Software engineering models and methods	MM
Software quality	SQ
Software engineering professional practice	PP
Software engineering economics	EE
Computing foundations	CF
Mathematical foundations	MF
Engineering foundations	EF

TABLE II. SWEBOK RELATED DISCIPLINES

Related Discipline	Abbreviation
Computer engineering	CE
Systems engineering	SY
Project management	PM
Quality management	QM
General management	GM

This paper, extending [1] with further details and incorporating an ontological perspective, contributes an SE edutainment approach we call SWE4SE for gamifying and conveying SE concepts to high school students. It describes SWE4SE principles, example mappings of SWEBOK concepts onto digital game (DG) elements, its realization, and evaluates its viability with an empirical study. It also investigates the feasibility of a methodological mapping of SE concepts to game element and game logic. Towards supporting efficient production of DGs for SE, we sought a viable method or platform for finding and using DGs for SE, or at least some support for (semi-)automatic mapping of SE concepts to DGs exists. Research questions investigated included: Is there clarity in the SE research or educational community as to which SE concepts can be effectively conveyed using which DG genre and game element types? Is this area mature, or is it subject to trial-and-error approaches? What sources of information or ontologies exist for this type of information? Where information is available, is this knowledge described more abstractly or formally to further reuse (e.g., for different DG genres or via standardized ontology formats)?

The paper is structured as follows: Section II provides background material about DGs and reviews literature on the application of digital game-based learning (DGBL) in SE education (SEE). Section III describes related work and Section IV the SWE4SE concept, followed by its realization. Section VI details the evaluation, and this is followed by a discussion. Section VIII provides a conclusion and describes future work.

II. BACKGROUND

This section provides background material about DG and reviews literature on the application of digital game-based learning (DGBL) in SE education (SEE).

A. DG Genres

According to [9], video game genres include Abstract, Adaptation, Adventure, Artificial Life, Board Games, Capturing, Card Games, Catching, Chase, Collecting, Combat, Demo, Diagnostic, Dodging, Driving, Educational, Escape, Fighting, Flying, Gambling, Interactive Movie, Management Simulation, Maze, Obstacle Course, Pencil-and-Paper Games, Pinball, Platform (levels and locomotion, e.g. Donkey Kong), Programming Games, Puzzle, Quiz, Racing, Role-Playing, Rhythm and Dance, Shoot 'Em Up, Simulation, Sports, Strategy, Table-Top Games, Target, Text Adventure, Training Simulation, and Utility.

B. Digital Game-based Learning (DGBL)

As argued in [14], DGBL research has largely shown that it is now accepted that DG can be an effective learning tool . However, further research is needed to explain why DGBL is effective and engaging, and "practical guidance for how (when, with whom, and under what conditions) games can be integrated into the learning process to maximize their learning potential." We lack research-supported guidance as to how and why DGBL is effective and how to actually implement DGBL. Conversely, [15] did not find strong evidence that games lead to more effective learning.

C. Motivational Aspects to Gaming

[15] found that users liked game-based learning and found it motivating and enjoyable. According to the Fogg Behavior Model [16], the motivation and the ability to perform must converge at the same moment for a behavior to occur. In the context of motivation in gaming, for the user to remain interested in the game positive feedback from game mechanics should continuously trigger a user to perform specific actions that the user has the ability to invoke. As such, any DGs targeted for secondary education students should appropriate for their knowledge and ability.

D. Game Ontology and Description

The goal of the Game Ontology Project (GOP) [17][18][19] is an ontology framework for describing, analyzing, and studying the design space of games with the

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

focus on gameplay. Its top-level concepts are Goals, Interfaces, Rules, Entity Manipulation, and Entities, whereby Entities are not further developed. GOP borrows concepts and methods from prototype theory [20] as well as grounded theory [21]. In contrast to gaming patterns [22], it provides the ability to identify and describe key structural elements rather than focusing on well-known solutions to recurring game design problems that have been codified as patterns. GOP concepts will be used to describe the realization of various games; however, it lacks the specification of genre and game logic. Other work on game ontologies includes [23], who investigated using Resource Description Framework (RDF) and Web Ontology Language (OWL) models to represent the knowledge within a role-playing DG.

[24] proposed the Video Game Description Language (VGDL) to be able to classify and describe the game logic, features, and mechanics of a game for the purposes of human and software agent understanding, as well as for automatic game generation from such a description. They focused on arcade games.

E. Literature Review of DGs Applied to SE

Rather than using the SWEBOK KAs, [13] used and extended the ISO/IEC 12207 to classify the SE process areas that were gamified in certain studies. Most of them focus on SD, and to a lesser extent SR, PM, and other support areas. Gamification elements employed by the primary studies were primarily points (14), badges (7), rankings (4), social reputation (4), voting (3), levels (2), visual metaphor (2), and otherwise quests, dashboard, betting, and awards.

Not all of the aforementioned studies explicitly differentiate on the axis of digital game (DG) from other serious game types, nor differentiate serious vs. fun games. Also, we noted that sometimes games were classified as pertaining to project management, but the work did not specifically focus on software project management.

Table III provides a summary of certain DG (based primarily on [13][25][26]) listing its genre with our own interpretation of the SWEBOK KA or SE-related concepts conveyed by various SE DG, extended with additional games. Our interpretation of the KA involved in certain games in [25] and [26] differ. If a paper provided no screenshot and no further evidence was found, then it is assumed not necessarily to be a digital game and was not included. If the game is not specifically focused on SE, then it was excluded. Note that while the [13] study appears to broadly cover SE gamification, for our purposes only 3 of the 29 cited primary papers, namely Trogon, HALO, and iThink, actually deal with digital games (have screenshots) that are focused on topics linked somewhat with SE.

[15] found that most common game genres in research were simulations (43), action games (14), puzzles (11), role-playing (8), strategy (6), and adventure (5). Only 1 of the 129 papers (that of Papastergiou) addressed a computing subject directly and only 3 of the 43 simulation game types were focused on entertainment as an intention, the rest were learning or serious games. This indicates that the simulation genre is most widespread in game research but entertainment is an uncommon intention for this genre. This correlates with our findings in Table III that simulation is a popular SE game genre in literature, and the likely intention of these games is primarily learning and not entertainment.

TABLE III. SELECTED EXAMPLES OF DIGITAL GAMES AND THE SE CONCEPTS THEY ATTEMPT TO CONVEY

Digital Game	Reference	Genre	SE Concepts Conveyed
Serious-AV (AVuSG)	Shabanah [27]	Simulation	CF
SimjavaSP	Shaw & Dermoudy [28]	Simulation	EM; PM
The Incredible Manager	Barros et al. [29]	Simulation	EM; PM
SimVBSE	Jain & Boehm [30]	Simulation	EM; PM
SimSoft	Caulfield et al. [31]	Simulation	EM; PM
Therefore iManage	Collofello [32]	Simulation	EM; EP; PM; EP
SESAM (Software Engineering Simulation by Animated Models)	Drappa & Ludewig [33]	Simulation	EM; EP; PM; EP
SimSE	Navarro & van der Hoek [34]	Simulation	EM; PM; EP; SR; EP; EM; PM
DesigMPS	Chaves et al. [35]	Simulation	EP
MO-SEProcess (SimSE in Second Life)	Wang & Zhu [36]	Simulation & Virtual Reality	EP; EM; PM
Trogon	Ašeriškis & Damaševičius [37]	Collaborative Simulation	PM
Pex4Fun	Xie et al. [38]	Social interactive coding	SC
-	Knauss et al. [39]	Simulation	SR
-	Hainey et al. [40]	Collaborative, Avatar	SR
iThink	Fernandes [41]	Simulation	SR
CIRCE, Production Cell, SummerSun, Quality Certification	Sharp & Hall [42]	Simulation	SR; SD; SC; SQ
HALO	Bell et al. [43]	Social quest	SD, ST

Unlisted are the plethora of software programming educational games or tools focused on teaching programming to kids, such as CodeCombat, CodeMonkey, CodinGame, Swift Playgrounds, Scratch, Alice, etc. However, our intent is not to teach a skill such as programming but to create an understanding and awareness about a SE topic. One reason for this is that we consider learning a skill (rather than learning about a skill) as requiring significantly more time investment for a DG user. Furthermore, programming overlaps with CS and does not help us to differentiate SE from CS.

Table III also shows that the most common KAs (10 of the 16) games listed are EM, PM, or EP and that a broader coverage of the 15 SWEBOK KAs with DGs is not apparent, be it any single game nor across multiple games.

F. SE Ontology Concepts

[44] and [45] provide an overview of the usage of various ontologies in SE.

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Work on SWEBOK ontologies in particular include [46], which performed an almost literal transcription, identifying over 4000 concepts, 400 relationships, 1200 facts, and 15 principles. While other work has also been done on a SWEBOK ontology, we were unable to access a any substantial SWEBOK ontology.

As far as utilizing the SWEBOK for SE education, [47] discusses modeling such an ontology.

III. Related Work

Beyond the related work in Section II, studies concerning the perception and attractiveness among secondary students of choosing CS as a college major, the study by [48] 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. The number one positive influence towards a major in CS for males was interest in computer games and for females, gaming was third. Among females, the primary positive motivator was the desire to use CS in another field, while this factor was third for males.

Serious games [12] have an explicit educational focus and tend to simulate real-world situations with intended audiences beyond secondary education. With regard to the use of gaming within the SE education field, [25] 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. A systematic survey on SE games by [26] analyzed 36 papers, all of which had targeted primarily undergraduate or graduates. A more recent study [13] carried out a systematic mapping to characterize the state of SE gamification, analyzing 29 primary studies from 2011-2014. It concluded that the and the application of gamification in SE is still in an initial stage, research in this area is quite preliminary, there is little sound evidence of its impact, and scarce empirical evidence.

Approaches for creating DGs include SimSYS [49], a model-driven engineering (MDE) based approach that integrates traditional entertainment game design elements, pedagogical content, and software engineering methodologies. Starting with informal models, it organizes games by acts, scenes, screens, and challenges and, using IBM Rhapsody, generates formal UML executable state chart models and XML for the SimSYS gameplay engine. However, this work does not describe which SE concepts map well to which game genres or game elements, nor does it attempt to utilize ontologies. [27] details game specifications, game genres, and game design, but is focused on the relatively narrow area of algorithm learning and visualization.

We were unable to find work related to the combination of SE or SWEBOK and DG ontologies or any more concrete method or mapping of SE concepts to game elements or game logic.

SWE4SE is targeted not towards higher education, but rather secondary school students with an explicit non-serious game approach. In secondary education, whereas initiatives for teaching programming are more common, conveying SE concepts in general and gamifying SE as a non-serious games has not hitherto been extensively studied, nor has the educational value of explicitly "non-serious" (or fun) games for this population stratum. While we study a secondary education population as does [48], our results 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. Furthermore, it attempts to explicitly describe the mapping of SE concepts to various DGs and game elements.

IV. SWE4SE Concept

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.

Based on the motivational aspect of gaming discussed in Section II, since our target behavior is interest in the subject matter of SE, for our target audience of secondary students lacking SE knowledge a typical SE DG that requires pre-knowledge of SE concepts will not motivate since they are lacking the skills to achieve the target behavior. This implies a SEE DG for our target audience should avoid the direct use of SE concepts in order to play the game.

A. 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

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

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 [50]. 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 control (SQC) game. This makes it more transparent how choices and actions are reflected in SE concepts.

Lower Bloom's Taxonomy Focus (P:BT): due to the limited time and attention for conveying SE concepts in the secondary school environment, the DGBL SE content and questions focus primarily on the lower levels of the cognitive domain: remembering (in the revised Bloom taxonomy [51]) or knowledge and comprehension (in the original Bloom taxonomy [52]). Note that the older 2004 version of the SWEBOK utilized the Bloom taxonomy to classify its knowledge content for educational purposes.

The aforementioned solution design principles are summarized in Table IV.

TABLE IV. SUMMARY OF SOLUTION DESIGN PRINCIPLES

Principle	Abbrev.
Web-browser Principle	P:Web
Engagement / Enjoyment Principle	P:En
Game Reuse Principle	P:GR
Simple Game Principle	P:SG
SE Concept Reinforcement via Game Action Principle	P:GA
Lower Bloom's Taxonomy Focus	P:BT

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 the 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, as summarized in Table V and detailed in Section V.

TABLE V. SE CONCEPT TO GAME MAPPING

SE Concept	SWEBOK KA	SWE4SE Game Variant	Analogous Common Game
Processes	EP, SC	ProcMan	Pac-Man
Quality control	SQ	Q-Check	Pinball
Requirements	SR	ReqAbdeck	Tower Defense
Testing	ST, SD	Angry Nerds	Angry birds
Construction	SC	Reverse Angry Nerds	Angry birds
Defect remediation	SM, CF	Bug Invaders	Space invaders
Project management	EM, PM	Path Management	Maze

The mapping should be interpreted as exploratory and not prescriptive; our intention here is rather to demonstrate the possibilities for conveying SE concepts such an edutainment approach provides.

V. SWE4SE REALIZATION

To develop the web-based games, Scirra Construct 2 was used, an HTML5 and JavaScript 2D game visual editor with a behavior-based logic system. 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 because we did not want language barriers for secondary students to affect our evaluation results, but the game text could be readily internationalized.

In the following description of each game, the entire upper levels of the SWEBOK ontology of the KA are also provided so that the actual SE concepts conveyed can be considered in context relative to other topics that were not conveyed. Since there was significant overlap in the GOP ontological game concepts used among the various games, these are presented at the end of the section.

A. Conveying SE Concepts in the Various Games

For each game, we describe how the analogous common games and their game concepts were mapped to corresponding SE concepts.

1) ***ProcMan****:* this game is analogous to the well-known Pac-Man game (see Figure 1), which has the highest brand awareness of U.S. consumers (94%) of any video game character [53].

a) Game Play: As in Pac-Man, the player controls the ProcMan traveling within a maze. Points are scored by eating the yellow pac-dots and bonus points are given if the cherries are eaten while shown just below the center. Starting with three lives, if during play one of the four chasing ghosts manages to touch the ProcMan a life is lost.

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

In our variant there is 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 having ProcMan consume the distributed initial letter standing for each phase in the expected order while also avoiding the enemy ghosts.

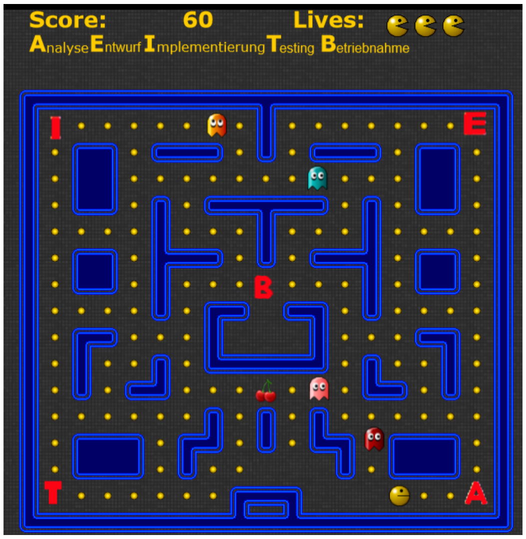

Figure 1. ProcMan game conveys SE processes (screenshot).

b) SE Concept: SE Processes. To convey an engineering process, we chose to introduce the activities common to almost any SE process. Based on the sequential 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).

c) SE Ontology Elements: within the EP KA, the SE concepts of Software Life Cycle Models and Software Process Adaptation are conveyed (see Figure 2).

- Software Engineering Process
  - Software Process Definition
    - Software Process Management
    - Software Process Infrastructure
  - Software Life Cycles
    - Categories of Software Processes
    - **Software Life Cycle Models**
    - **Software Process Adaptation**
    - Practical Considerations
  - Software Process Assessment and Improvement
    - Software Process Assessment Models
    - Software Process Assessment Methods
    - Software Process Improvement Models
    - Continuous and Staged Software Process Ratings
  - Software Measurement
    - Software Process and Product Measurement
    - Quality of Measurement Results
    - Software Information Models
    - Software Process Measurement Techniques
  - Software Engineering Process Tools

Figure 2. SWEBOK EP KA concepts (in bold) conveyed by ProcMan.

Within the SC KA, the SE concept of Test-First Programming is conveyed (see Figure 3).

- Software Construction
  - Software Construction Fundamentals
    - Minimizing Complexity
    - Anticipating Change
    - Constructing for Verification
    - Reuse
    - Standards in Construction
  - Managing Construction
    - Construction in Life Cycle Models
    - Construction Planning
    - Construction Measurement
  - Practical Considerations
    - Construction Design
    - Construction Languages
    - Coding
    - Construction Testing
    - Construction for Reuse
    - Construction with Reuse
    - Construction Quality
    - Integration
  - Construction Technologies
    - API Design and Use
    - Object-Oriented Runtime Issues
    - Parameterization and Generics
    - Assertions, Design by Contract, and Defensive Programming
    - Error Handling, Exception Handling, and Fault Tolerance
    - Executable Models
    - State-Based and Table-Driven Construction Techniques
    - Runtime Configuration and Internationalization
    - Grammar-Based Input Processing
    - Concurrency Primitives
    - Middleware
    - Construction Methods for Distributed Software
    - Constructing Heterogeneous Systems
    - Performance Analysis and Tuning
    - Platform Standards
    - **Test-First Programming**
  - Software Construction Tools
    - Development Environments
    - GUI Builders
    - Unit Testing Tools
    - Profiling, Performance Analysis, and Slicing Tools

Figure 3. SWEBOK SC KA concepts (in bold) conveyed by ProcMan.

2) Q-Check: this game is loosely analogous to pinball (see Figure 4).

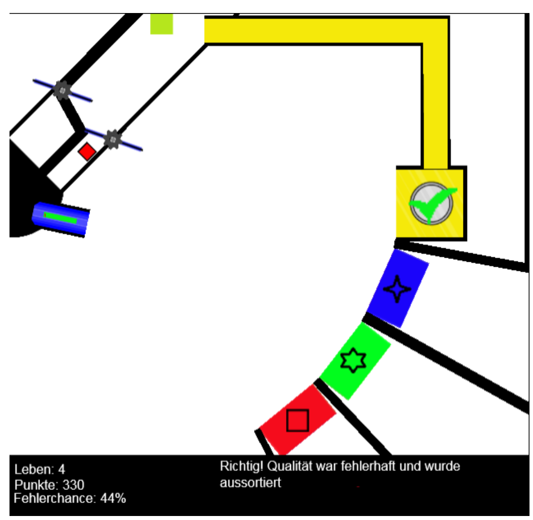

Figure 4. Q-Check game conveys SE quality control (screenshot).

a) Game Play: 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.

b) SE Concept: Software quality control (SQC). Quality expectations differ based on the type of software component being inspected (e.g., GUI, database, business logic). Quality awareness and attention to detail matter, and the appropriate quality process, tools, and testing procedures must be chosen dependent on the assessed object.

c) SE Ontology Elements: within the SQ KA, the SE concept of Software Quality Management Processes is conveyed as shown in Figure 5.

- Software Quality
  - Software Quality Fundamentals
    - Software Engineering Culture and Ethics
    - Value and Costs of Quality
    - Models and Quality Characteristics
    - Software Quality Improvement
    - Software Safety
  - **Software Quality Management Processes**
    - Software Quality Assurance
    - Verification & Validation
    - Reviews and Audits
  - Practical Considerations
    - Software Quality Requirements
    - Defect Characterization
    - Software Quality Management Techniques
    - Software Quality Measurement
  - Software Quality Tools

Figure 5. SWEBOK concepts (in bold) conveyed by Q-Check.

3) ***ReqAbdeck:*** ("Abdeckung" in German means "coverage") this game is analogous to the popular game Tower Defense (see Figure 6).

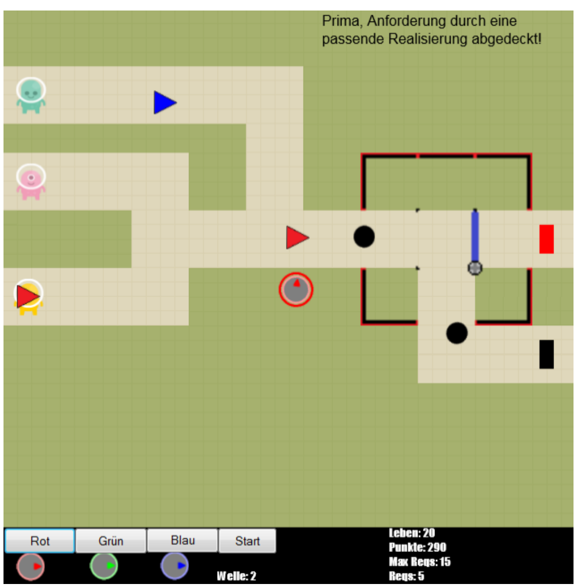

Figure 6. ReqAbdeck conveys SE requirement coverage (screenshot).

a) Game Play: 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.

b) SE Concept: Software requirements. ReqAbdeck 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).

c) SE Ontology Elements: within the SR KA, the SE concepts of Requirements Process Quality and Improvement and Requirements Classification are conveyed as shown in Figure 7.

- Software Requirements
  - Software Requirements Fundamentals
    - Definition of a Software Requirement
    - Product and Process Requirements
    - Functional and Nonfunctional Requirements
    - Emergent Properties
    - Quantifiable Requirements
    - System Requirements and Software Requirements
  - Requirements Process
    - Process Models
    - Process Actors
    - Process Support and Management
    - **Process Quality and Improvement**
  - Requirements Elicitation
    - Requirements Sources
    - Elicitation Techniques
  - Requirements Analysis
    - **Requirements Classification**
    - Conceptual Modeling
    - Architectural Design and Requirements Allocation
    - Requirements Negotiation
    - Formal Analysis
  - Requirements Specification
    - System Definition Document
    - System Requirements Specification
    - Software Requirements Specification
  - Requirements Validation
    - Requirements Reviews
    - Prototyping
    - Model Validation
    - Acceptance Tests
  - Practical Considerations
    - Iterative Nature of the Requirements Process
    - Change Management
    - Requirements Attributes
    - Requirements Tracing
    - Measuring Requirements
  - Software Requirements Tools

Figure 7. SWEBOK concepts (in bold) conveyed by ReqAbdeck.

4) ***Angry Nerds:*** this game is loosely analogous to the popular game Angry Birds (see Figure 8).

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

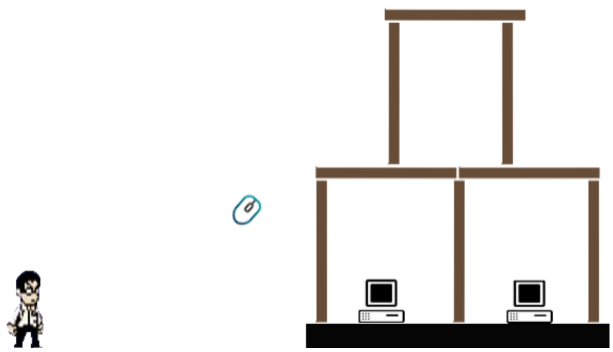

Figure 8. AngryNerds conveys SE testing (screenshot).

a) Game Play: we chose to depict hardware-like testing here of children's blocks, since it was not obvious to us how to quickly convey code-based testing in an obvious manner without necessitating extra explanations about programming. 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.

b) SE Concept: Software testing. The SE focus of this game is to convey the objectives of testing (finding deficiencies and building confidence in one's construct) determining where to test to find deficiencies in some software construct.

c) SE Ontology Elements: within the ST KA, the SE concept of Objectives of Testing is conveyed as shown in Figure 9.

- Software Testing
  - Software Testing Fundamentals
    - Testing-Related Terminology
    - Key Issues
    - Relationship of Testing to Other Activities
  - Test Levels
    - The Target of the Test
    - **Objectives of Testing**
  - Test Techniques
    - Based on the Software Engineer's Intuition and Experience
    - Input Domain-Based Techniques
    - Code-Based Techniques
    - Fault-Based Techniques
    - Usage-Based Techniques
    - Model-Based Testing Techniques
    - Techniques Based on the Nature of the Application
    - Selecting and Combining Techniques
  - Test-Related Measures
    - Evaluation of the Program Under Test
    - Evaluation of the Tests Performed
  - Test Process
    - Practical Considerations
    - Test Activities
  - Software Testing Tools
    - Testing Tool Support
    - Categories of Tools

Figure 9. SWEBOK ST concepts (in bold) conveyed by Angry Nerds.

Within the KA SD, the SE concept of Quality Analysis and Evaluation Techniques is conveyed as shown in Figure 10.

- Software Design
  - Software Design Fundamentals
    - General Design Concepts
    - Context of Software Design
    - Software Design Process
    - Software Design Principles
  - Key Issues in Software Design
    - Concurrency
    - Control and Handling of Events
    - Data Persistence
    - Distribution of Components
    - Error and Exception Handling and Fault Tolerance
    - Interaction and Presentation
    - Security
  - Software Structure and Architecture
    - Architectural Structures and Viewpoints
    - Architectural Styles
    - Design Patterns
    - Architecture Design Decisions
    - Families of Programs and Frameworks
  - User Interface Design
    - General User Interface Design Principles
    - User Interface Design Issues
    - The Design of User Interaction Modalities
    - The Design of Information Presentation
    - User Interface Design Process
    - Localization and Internationalization
    - Metaphors and Conceptual Models
  - Software Design Quality Analysis and Evaluation
    - Quality Attributes
    - **Quality Analysis and Evaluation Techniques**
    - Measures
  - Software Design Notations
    - Structural Descriptions (Static View)
    - Behavioral Descriptions (Dynamic View)
  - Software Design Strategies and Methods
    - General Strategies
    - Function-Oriented (Structured) Design
    - Object-Oriented Design
    - Data Structure-Centered Design
    - Component-Based Design (CBD)
    - Other Methods
  - Software Design Tools

Figure 10. SWEBOK SD concepts (in bold) conveyed by Angry Nerds.

5) ***Reverse Angry Nerds****:* this game actually becomes available in the bonus level of the previous game, but the gameplay is different, in that it reverses the role as shown in Figure 11.

Figure 11. Reverse AngryNerds game conveys SE construction (screenshot).

a) Game Play: the player must now try to build a resilient construct by dragging and placing slabs in such a way that it withstands the automated testing (that being a cannonball shot at the construct).

b) SE Concept: Software construction. The point of this exercise is to construct something (analogous to software) such that it exhibits resiliency.

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

c) SE Ontology Elements: within the ST KA, the SE concept of Construction Design is conveyed as shown in Figure 12.

- Software Construction
  - Software Construction Fundamentals
    - Minimizing Complexity
    - Anticipating Change
    - Constructing for Verification
    - Reuse
    - Standards in Construction
  - Managing Construction
    - Construction in Life Cycle Models
    - Construction Planning
    - Construction Measurement
  - Practical Considerations
    - **Construction Design**
    - Construction Languages
    - Coding
    - Construction Testing
    - Construction for Reuse
    - Construction with Reuse
    - Construction Quality
    - Integration
  - Construction Technologies
    - API Design and Use
    - Object-Oriented Runtime Issues
    - Parameterization and Generics
    - Assertions, Design by Contract, and Defensive Programming
    - Error Handling, Exception Handling, and Fault Tolerance
    - Executable Models
    - State-Based and Table-Driven Construction Techniques
    - Runtime Configuration and Internationalization
    - Grammar-Based Input Processing
    - Concurrency Primitives
    - Middleware
    - Construction Methods for Distributed Software
    - Constructing Heterogeneous Systems
    - Performance Analysis and Tuning
    - Platform Standards
    - Test-First Programming
  - Software Construction Tools
    - Development Environments
    - GUI Builders
    - Unit Testing Tools
    - Profiling, Performance Analysis, and Slicing Tools

Figure 12. SWEBOK concepts (in bold) conveyed by Angry Nerds.

*6) **Bug Invaders**:* this game is analogous to space invaders, see Figure 13.

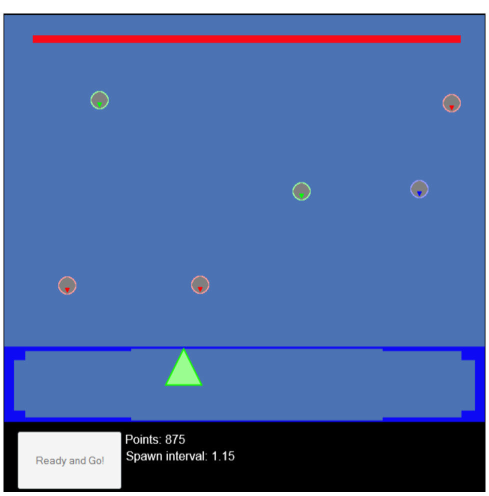

Figure 13. Bug Invaders convey SE defect remediation (screenshot).

a) Game Play: a matching remediation technique (maps to ammunition color in the lower shooter) and firing accuracy (maps to exact causal code location) are needed to destroy exactly that specific bug type that drops down quickly before it creates project damage.

b) SE Concept: Software defect remediation. The SE focus of this game is to convey that different defect types require different remediation techniques and countermeasures applied accurately.

c) SE Ontology Elements: within the KA SM, the SE concept of Maintenance Activities is conveyed as shown in Figure 14.

- Software Maintenance
  - Software Maintenance Fundamentals
    - Definitions and Terminology
    - Nature of Maintenance
    - Need for Maintenance
    - Majority of Maintenance Costs
    - Evolution of Software
    - Categories of Maintenance
  - Key Issues in Software Maintenance
    - Technical Issues
    - Management Issues
    - Maintenance Cost Estimation
    - Software Maintenance Measurement
  - Maintenance Process
    - Maintenance Processes
    - **Maintenance Activities**
  - Techniques for Maintenance
    - Program Comprehension
    - Reengineering
    - Reverse Engineering
    - Migration
    - Retirement
  - Software Maintenance Tools

Figure 14. SWEBOK SM concepts (in bold) conveyed by Bug Invaders.

Within the CF KA, the SE concepts of Types of Errors and Debugging Techniques are conveyed as shown in Figure 16.

*7) **Path Management**:* this game is analogous to a maze, see Figure 15.

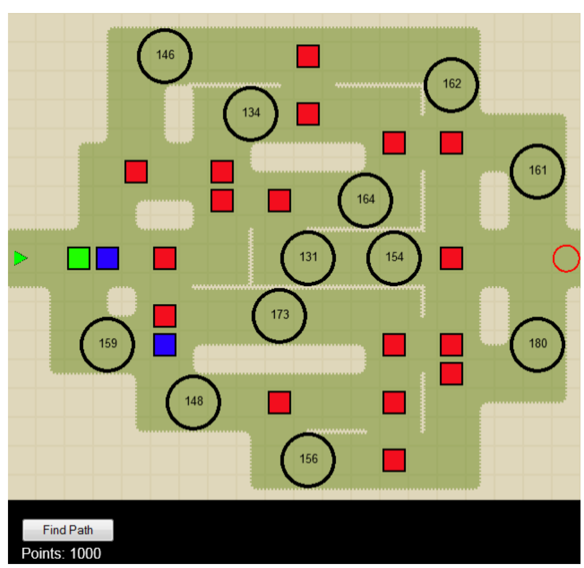

Figure 15. Path Management conveys SE project management (screenshot).

- Computing Foundations
  - Problem Solving Techniques
    - Definition of Problem Solving
    - Formulating the Real Problem
    - Analyze the Problem
    - Design a Solution Search Strategy
    - Problem Solving Using Programs
  - Abstraction
    - Levels of Abstraction
    - Encapsulation
    - Hierarchy
    - Alternate Abstractions
  - Programming Fundamentals
    - The Programming Process
    - Programming Paradigms
  - Programming Language Basics
    - Programming Language Overview
    - Syntax and Semantics of Programming Languages
    - Low-Level Programming Languages
    - High-Level Programming Languages
    - Declarative v+ Imperative Programming Languages
  - Debugging Tools and Techniques
    - **Types of Errors**
    - **Debugging Techniques**
    - Debugging Tools
  - Data Structure and Representation
    - Data Structure Overview
    - Types of Data Structure
    - Operations on Data Structures
  - Algorithms and Complexity
    - Overview of Algorithms
    - Attributes of Algorithms
    - Algorithmic Analysis
    - Algorithmic Design Strategies
    - Algorithmic Analysis Strategies
  - Basic Concept of a System
    - Emergent System Properties
    - Systems Engineering
    - Overview of a Computer System
  - Computer Organization
    - Computer Organization Overview
    - Digital Systems
    - Digital Logic
    - Computer Expression of Data
    - The Central Processing Unit (CPU)
    - Memory System Organization
    - Input and Output (I/O)
  - Compiler Basics
    - Compiler/Interpreter Overview
    - Interpretation and Compilation
    - The Compilation Process
  - Operating Systems Basics
    - Operating Systems Overview
    - Tasks of an Operating System
    - Operating System Abstractions
    - Operating Systems Classification
  - Database Basics and Data Management
    - Entity and Schema
    - Database Management Systems (DBMS)
    - Database Query Language
    - Tasks of DBMS Packages
    - Data Management
    - Data Mining
  - Network Communication Basics
    - Types of Network
    - Basic Network Components
    - Networking Protocols and Standards
    - The Internet
    - Internet of Things
    - Virtual Private Network (VPN)
  - Parallel and Distributed Computing
    - Parallel and Distributed Computing Overview
    - Difference between Parallel and Distributed Computing
    - Parallel and Distributed Computing Models
    - Main Issues in Distributed Computing
  - Basic User Human Factors
    - Input and Output
    - Error Messages
    - Software Robustness
  - Basic Developer Human Factors
    - Structure
    - Comments
  - Secure Software Development and Maintenance
    - Software Requirements Security
    - Software Design Security
    - Software Construction Security
    - Software Testing Security
    - Build Security into Software Engineering Process
    - Software Security Guidelines

Figure 16. SWEBOK CF concepts (in bold) conveyed by Bug Invaders.

a) Game Play:.a player must manage a starting budget in points efficiently. 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.

b) SE Concept: software project management. The SE concept conveyed is that multiple choices towards optimizing project costs exist, and the process is planned and resources allocated considering various risks. With appropriate planning, the project goal can be reached with the allotted resources, despite unexpected problems (risk transition) that must be overcome but nevertheless result in unplanned additional resource costs.

c) SE Ontology Elements: within the KA of EM, the concepts of Process Planning;, Effort, Schedule, and Cost Estimation; Resource Allocation; and Risk Management are conveyed, see Figure 17.

- Software Engineering Management
  - Initiation and Scope Definition
    - Determination and Negotiation of Requirements
    - Feasibility Analysis
    - Process for the Review and Revision of Requirements
  - Software Project Planning
    - **Process Planning**
    - Determine Deliverables
    - **Effort, Schedule, and Cost Estimation**
    - **Resource Allocation**
    - **Risk Management**
    - Quality Management
    - Plan Management
  - Software Project Enactment
    - Implementation of Plans
    - Software Acquisition and Supplier Contract Management
    - Implementation of Measurement Process
    - Monitor Process
    - Control Process
    - Reporting
  - Review and Evaluation
    - Determining Satisfaction of Requirements
    - Reviewing and Evaluating Performance
  - Closure
    - Determining Closure
    - Closure Activities
  - Software Engineering Measurement
    - Establish and Sustain Measurement Commitment
    - Plan the Measurement Process
    - Perform the Measurement Process
    - Evaluate Measurement
  - Software Engineering Management Tools

Figure 17. SWEBOK concepts (in bold) conveyed by Path Management.

B. Realization of the Game World SE Exploration Concept

To tie the various DGs together, the realization includes a SE universe to navigate to and discover various SE planets. Figure 18 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 19), a short text about SE concepts that relates to the game is shown, which when

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

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.

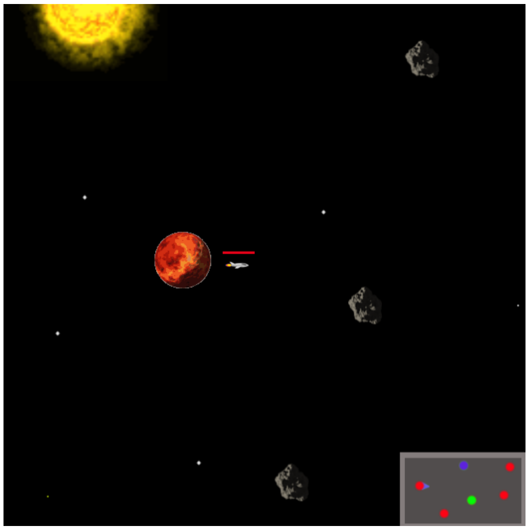

Figure 18. Spaceship navigating the SE universe (screenshot).

Figure 19. 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).

C. Game Ontology

Instantiated GOP game ontology concepts are marked with italics in Figure 20, with constraints or notes provided in parentheses. Additional unutilized GOP leaf nodes are not depicted due to space limitations.

GOP
- Interface
  - Input
    - Input Devices
      - *Keyboard*
      - *Mouse*
  - Input Method
    - Locus of Manipulation
      - *Single Entity Manipulation*
    - Manipulation Method
      - Indirect Manipulation
        - *Point and Click Interface*
  - Presentation
    - Presentation Hardware
      - Visual Display Hardware
        - *Video Monitor*
    - Cardinality Of Gameworld
      - *2-Dimensional Gameworld*
    - Presentation Software
      - Sensory Output
        - Visual Output
          - Head Up Display
            - *Lives Indicator* (except PathManagement)
            - *Points Indicator*
            - *Health Indicator*
            - *Map Display* (Universe)
            - *Next Piece Indicator* (Q-Check)
        - *Aural Output* (music)
      - Point of View
        - *Second-person Point of View*
      - Feedback
        - *Confirmation* (messages, badges)
- Entity Manipulation
  - *To Move*
  - *To Remove* (except Path Management, Reverse AngryNerds, Universe)
  - *To Rotate* (ReqAbdeck, Universe, Q-Check)
  - *To Collide* (ProcMan, Universe)
  - *To Select*
  - Compound Action
    - *To Shoot* (AngryNerds)
    - *To Manage Resources* (Reverse AngryNerds)
- Rules
  - Gameworld Rules
    - Pseudo-Physical Rules
      - *Solidity* (AngryNerds, Reverse AngryNerds)
      - *Gravity* (AngryNerds, Reverse AngryNerds, Universe)
  - Gameplay Rules
    - Cardinality of Gameplay
      - *0D Gameplay* (Q-Check, AngryNerds, Reverse AngryNerds)
      - *1D Gameplay* (Bug Invaders)
      - *2D Gameplay* (ProcMan, ReqAbdeck, Universe)
    - *Lives*
    - *Randomness*
    - Game Ends
      - *Resource Exhaustion* (if lives lost)
      - *Gameworld Exhaustion* (some max. time)
      - *No Game End* (Universe)
    - Segmentation of Gameplay
      - Spatial Segmentation
        - *Level* (advanced planet areas)
      - Challenge Segmentation
        - *Wave* (ReqAbdeck, Bug Invaders)
        - *Puzzle* (PathManagement)
        - *Bonus Stage*
      - Temporal Segmentation
        - *Temporal Resource* (time limits)
    - Rules Synergies
      - *Complete Information* (except PathManagement)
      - *Dynamic Difficulty Adjustment* (Q-Check, ReqAbdeck, Bug Invaders)
    - *Spawnpoint*
- Goals
  - Game Goals
    - *Required Goals*
  - Goal Metrics
    - *Score*

Figure 20. GOP concepts (in italics) utilized by SWE4SE.

D. Realization via Game Framework

Construct 2 was used to realize the games. Generally, the browser must support HTML5 and Javascript and popups must be activated. Scirra Ltd. recommends Mozilla Firefox, Google Chrome, or Opera. Internet Explorer 9+ can be used from Windows Vista on, Windows XP cannot used IE. We had problems with sound using IE (11 was tested).

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

VI. SWE4SE Evaluation

The convenience sampling technique [54], 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.

A. Evaluation Setting

Two teachers at two different public university preparatory (secondary) schools in different cities in the local German region gave us access for 90 minutes to 20 total students that attend their informatics interest groups. *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 21 shows the age and gender distribution, and Figure 22 indicates their current game usage frequency.

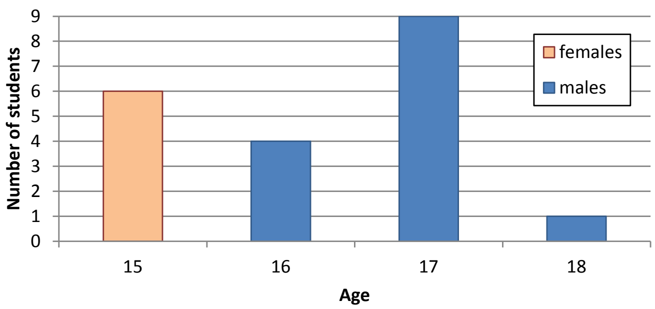

Figure 21. Student age and gender distribution.

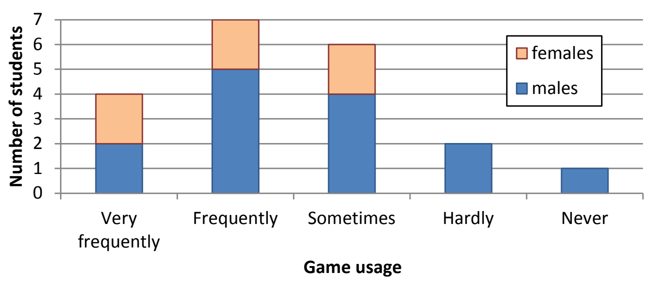

Figure 22. Prior game usage frequency distribution.

B. Evaluation Method

While we considered utilizing the GEQ [55], it appeared more appropriate for more immersive games rather than our game genres. Due to factors such as 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, or did not know how to answer, or did not care to answer the question. Blank answers were thus omitted.

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.

C. Evaluation Results

We observed that all students were engaged with SWE4SE for the entire hour and appeared to enjoy the experience *(P:En),* and occasionally interacted excitedly with fellow students. Table VI provides 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):

TABLE VI. User Experience

Factor	Rating	Relates to
Overall experience	2.1	*P:En*
Game enjoyment	2.0	*P:En*
Helpful conveying several SE concepts via different games	Yes (16) No (1)	*P:SG, P:GR*
Success rate recalling the SE concepts associated with each named game[a]	62%	*P:GA* Text Components
Watched the video attentively	Yes (20)	
Video and its quality	Good (20)	
Video length of 5 minutes	Keep (19) Lengthen (1)	

a. Open answers. 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.

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

TABLE VII. 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.

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

VII. DISCUSSION

We now discuss our evaluation results and findings.

A. Evaluation

Our evaluation showed that the SWE4SE approach can be effective: 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. 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.

B. Ontologies and Methods supporting DGBL in SE

In our investigation of which types of SE concepts lend themselves to being conveyed with which game genres, elements, and gameplay or logic we noticed various difficulties with regard to achieving clarity or more formalization as to a method for mapping SE educational concepts to game genres or elements. This was due to a number of reasons, such as a lack of additional information to more precisely categorize the concepts from various viewpoints, and lack of accessibility of standardized ontologies in standard formats.

For instance, while we were able to categorize the SWEBOK knowledge being conveyed at a relatively coarse granularity in its taxonomy, we found it to be missing additional ontological relations or properties. For one, Bloom's taxonomy has now been removed from the SWEBOK, making it more difficult to automatically or quickly assess the type of knowledge. E.g., if we are only interested in the remembering level of knowledge we cannot easily cull this from the rest.

From the DG standpoint, we were able to describe the various low-level game elements with the GOP. However, we lacked the ability to describe the gameplay logic. For instance, does the game involve sequencing, or differentiating objects, constructing, destroying, analysis, or planning?

A method for mapping we could conceive of would be that any SE concept of the type *process* could perhaps be mapped to a DG that involves sequencing, or SE concept of the type *analysis* could utilize a DG that involves differentiation.

C. State of DGBL in SE

Our literature review of DG in SE did not find a broad coverage of KAs. For the purpose of conveying SE concepts specifically in the secondary education this may not be necessary, but it does perhaps indicate that DG have concentrated in certain more obvious areas and that other SE areas for applying DGs have perhaps not been sufficiently explored.

Other than programming games, there appears to be little commercial interest or incentive to provide professional DG to the SE educational community. It is our opinion that DGBL for SE is currently in a relatively immature state, that developed DGs are typically analogous to islands in an uncharted sea and not readily discoverable, accessible, and reusable. We thus suggest that the SE community develop a common SE DG platform that would provide convenient access to such DGs, broadening their discovery and reuse, supporting their open source evolution, and providing feedback. If the community had a categorization utilizing the SWEBOK or similar SE ontology and linked available DGs that address these, then DGBL reuse could be furthered for conveying SE concepts.

VIII. 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 from an ontological perspective how various SE concepts could be mapped and realized with various digital game concepts and elements.

As an indicator of the economic realizability of SWE4SE, our DG realization was done by two students in a 10 credit project in one semester, equivalent to approximately 600 hours workload.

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.

As to further development, refinement, and evolution of such an SWE4SE or similar game development approach, we believe ontologies to be promising for more formally conveying knowledge concepts for the SE domain and for describing various game concepts. However, our investigation determined that a severe gap and immaturity exists in this area that prevents the (semi-)automated inclusion and mapping of SE concepts to game objects or gameplay logic (e.g., via game description languages). This area should thus be further investigated, developed, and formalized to more effectively support DGBL and DG reuse and know-how for SEE and move this area from a trail-and-error experimental craft to more professional engineering.

Future work includes investigating the integration both game and SE domain ontologies into game engines and

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

description languages, a longitudinal study on motivational effect retention and other interfering or conflicting influences, and integrated game and web analytics to provide further insights into game playing behavior.

ACKNOWLEDGMENT

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

REFERENCES

[1] R. Oberhauser, "Gamifying and Conveying Software Engineering Concepts for Secondary Education: An Edutainment Approach," In: Proc. of the Tenth International Conf. on Software Engineering Advances (ICSEA 2015). IARIA XPS Press, 2015. pp. 432-437, ISBN: 978-1-61208-367-4.

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

[3] B. Schmidt, "How are college majors changing?" [Online]. Available from: http://benschmidt.org/Degrees/

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

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

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

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

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

[9] M. Wolf, The medium of the video game. University of Texas Press, 2002.

[10] I. Bogost, Persuasive games: The expressive power of videogames. The MIT Press, 2007.

[11] D.R. Michael and S.L. Chen, , Serious games: Games that educate, train, and inform. Muska & Lipman/Premier-Trade. 2005.

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

[13] O. Pedreira, F. García, N. Brisaboa, and M. Piattini,. "Gamification in software engineering–A systematic mapping," Information and Software Technology, 57, 2015, pp.157-168.

[14] Van Eck, R., 2006. Digital game-based learning: It's not just the digital natives who are restless. EDUCAUSE review, 41(2), p.16.

[15] T. M. Connolly, E. A. Boyle, E. MacArthur T. Hainey, J. M. Boyle, "A systematic literature review of empirical evidence on computer games and serious games," Computers & Education, 59, Elsevier Ltd., 2012, pp. 661–686.

[16] B. J. Fogg, "A behavior model for persuasive design," In Proceedings of the 4th international Conference on Persuasive Technology, ACM, 2009. pp. 40-46.

[17] The Game Ontology Project. [Online]. Available from: http://www.gameontology.com

[18] J. P. Zagal and A. Bruckman, "The game ontology project: supporting learning while contributing authentically to game studies," in Proc. 8th International conference for the learning sciences (ICLS '08), Vol. 2, International Society of the Learning Sciences, 2008, pp. 499-506.

[19] J.P. Zagal, Ludoliteracy: Defining, Understanding, and Supporting Games Education. ETC Press, 2011.

[20] G. Lakoff, Women, Fire, and Dangerous Things: What Categories Reveal About the Mind. The University of Chicago Press, Chicago, 1987.

[21] B. Glaser and A. Strauss, The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine, Chicago, 1967.

[22] D. Ašeriškis and R. Damaševičius, "Gamification Patterns for Gamification Applications," in Proc. of the 6th international conference on Intelligent Human Computer Interaction (IHCI 2014), Procedia Computer Science, Vol. 39, 2014, pp.83-90.

[23] J. T. Chan and W. Y. Yuen, "Digital Game Ontology: Semantic Web Approach on Enhancing Game Studies," In Proceedings of 2008 IEEE 9th International Conference on Computer-Aided Industrial Design & Conceptual Design, Vol. 1, 2008.

[24] M. Ebner et al., "Towards a video game description language," In: Artificial and Computational Intelligence in Games, Dagstuhl Follow-ups, 6. Dagstuhl Publishing, Wadern, 2013, pp. 85-100, ISBN 9783939897620.

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

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

[27] S. Shabanah, "Computer Games for Algorithm Learning," In Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches, ed. Patrick Felicia, 2011, pp. 1036-1063.

[28] K. Shaw and J. Dermoudy, "Engendering an Empathy for Software Engineering," In Proceedings of the 7th Australasian conference on Computing education-Volume 42. Australian Computer Society, Inc., 2005, pp. 135-144.

[29] M. d. O. Barros, A. R. Dantas, G. O. Veronese, and C. M. L. Werner, "Model-Driven Game Development: Experience and Model Enhancements in Software Project Management Education," in Software Process: Improvement and Practice. 11(4), 2006, pp. 411–421, doi:10.1002/spip.279.

[30] A. Jain and B. Boehm, "SimVBSE: Developing a Game for Value-Based Software Engineering," In Proceedings of the 19th Conference on Software Engineering Education & Training (CSEET'06), IEEE, 2006, pp. 103-114, doi:10.1109/cseet.2006.31.

[31] C. W. Caulfield, D. Veal, and S. P. Maj, "Teaching software engineering management–issues and perspectives," International Journal of Computer Science and Network Security, 11(7), 2011, pp.50-54.

[32] J. S. Collofello, "University/industry collaboration in developing a simulation-based software project management training course," IEEE Transactions on Education, 43(4), IEEE, 2000, pp.389-393.

[33] A. Drappa and J. Ludewig, "Simulation in software engineering training," in Proceedings of the 22nd International Conference on Software Engineering, 2000, pp. 199–208.

[34] E. O., Navarro and A. van der Hoek, "Multi-Site Evaluation of SimSE," Proceedings of The 40th ACM Technical Symposium on Computer Science Education, (SIGCSE '09). ACM, 2009, pp. 326-330, doi:10.1145/1508865.1508981.

[35] R. O. Chaves et al., "Experimental evaluation of a serious game for teaching software process modeling," IEEE Transactions on Education, 58(4), 2015, pp.289-296.

[36] T. Wang and Q. Zhu, "A Software Engineering Education Game in a 3-D Online Virtual Environment," Proceedings of The First International Workshop on Education Technology

2016, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

and Computer Science, Vol. 2. IEEE, 2009. pp. 708-710, doi:10.1109/ETCS.2009.418.

[37] D. Ašeriškis and R. Damaševičius, "Gamification of a project management system," in Proc. of Int. Conf. on Advances in Computer-Human Interactions (ACHI2014), 2014, pp. 200-207.

[38] T. Xie, N. Tillmann, and J. De Halleux, "Educational software engineering: Where software engineering, education, and gaming meet," In Proceedings of the 3rd International Workshop on Games and Software Engineering: Engineering Computer Games to Enable Positive, Progressive Change. IEEE Press, 2013, pp. 36-39.

[39] E. Knauss, K. Schneider, K., and K. Stapel, "A Game for Taking Requirements Engineering More Seriously," Proceedings of The Third International Workshop on Multimedia and Enjoyable Requirements Engineering - Beyond Mere Descriptions and with More Fun and Games. IEEE, 2008, pp. 22-26, doi:10.1109/MERE.2008.1.

[40] T. Hainey, T. J. Connelly, M. Stansfield, and E. A. Boyle, "Evaluation of a Game to Teach Requirements Collection and Analysis in Software Engineering at Tertiary Education Level," Computers & Education, 56(1), 2010, pp. 21–35, doi: 10.1016/j.compedu.2010.09.0.

[41] J. Fernandes et al., "iThink: A Game-Based Approach Towards Improving Collaboration and Participation in Requirement Elicitation," in Proc. of the 4th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES'12). Procedia Computer Science, vol. 15, 2012, pp. 66-77, doi:10.1016/j.procs.2012.10.059.

[42] H. Sharp and P. Hall, "An Interactive Multimedia Software House Simulation for Postgraduate Software Engineers," in Proceedings of the 22nd International Conference on Software Engineering. ACM, 2000, pp. 688-691.

[43] J. Bell, S. Sheth, S., and G. Kaiser, "Secret ninja testing with HALO software engineering," In Proceedings of the 4th international workshop on social software engineering. ACM, 2011, pp. 43-47.

[44] C. Calero, F. Ruiz, and M. Piattini, M. (eds.), Ontologies for software engineering and software technology. Springer Science & Business Media, 2006.

[45] M. P. S. Bhatia, A. Kumar, and R. Beniwal, "Ontologies for Software Engineering: Past, Present and Future," Indian Journal of Science and Technology, 9(9), 2016, pp. 1-16.

[46] O. Mendes and A. Abran, "Issues in the development of an ontology for an emerging engineering discipline," In Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering (SEKE'2005), 2005, pp. 139-144.

[47] C. Wille, R. R. Dumke, A. Abran, and J. M. Desharnais, "E-learning infrastructure for software engineering education: Steps in ontology modeling for SWEBOK," In IASTED International Conf. on Software Engineering, 2004, pp. 520-525.

[48] 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.

[49] K. M. Cooper and C. S. Longstreet, "Model-driven Engineering of Serious Educational Games: Integrating Learning Objectives for Subject Specific Topics and Transferable Skills," In Computer Games and Software Engineering (1st ed.), K. M. Cooper and W. Scacchi, Eds. Chapman & Hall/CRC, 2015, pp. 59-90.

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

[51] Anderson, L., Krathwohl, David R, & Bloom, Benjamin S. (2001). A taxonomy for learning, teaching, and assessing : A revision of Bloom's taxonomy of educational objectives. New York: Longman.

[52] Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; Krathwohl, D. R. (1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York: David McKay Company.

[53] Davie Brown Entertainment, "Davie Brown Celebrity Index: Mario, Pac-Man Most Appealing Video Game Characters Among Consumers". PR Newswire, 2008. [Online]. Available from: http://www.prnewswire.com/news-releases/davie-brown-celebrity-index-mario-pac-man-most-appealing-video-game-characters-among-consumers-57256317.html

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

[55] 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.