Technical Report TR-MiamiUniversity-CSE-2015-001

Gamification in SE Courses

Focused on Increased Motivation of Female Students

James D. Kiper
Department of Computer Science and Software Engineering
Miami University
Oxford OH, USA
James.Kiper@miamiOH.edu

Abstract—The sources and styles of student motivation have changed as the use of computing technology has become more ubiquitous. Instructors observations often lead to the conclusion that students' attention span has shorten. Nevertheless, these students often focus for long periods of times on games, both video games and board games. The thesis of this research is that gamification techniques can be used to increase students' motivation in learning various aspects of software engineering. Furthermore, by exploiting certain types of games, we hypothesize can increase the interest and motivation of female software engineering students. The research described herein is a work in progress to examine these hypotheses in the context of teaching in a specific area of software engineering: software testing.

Index Terms—Games, gamification, software engineering education, teaching software testing.

I. Introduction

One metric that can be an accurate predictor of success in learning is *time on task*. The research of K. Anders Ericsson on expertise [7] suggests that it takes 10,000 hours of deliberate practice to become an expert in any skill. Personal experience for most of us will validate the assertion that time wisely spent on a particular task is requisite for learning. There can be no doubt that many of our students have validated this hypothesis to game playing. Likewise, many of our colleagues have used this approach in becoming experts in games like chess.

However, there is a perception among many educators that time spent on games is one of the problems in education, not the solution. Those of us who teach early morning classes sometimes hypothesize that the sleepiness of students or their absence from these classes is the result of late night game playing. Can we dare think of game playing as something to encourage?

Specifically, the idea is to increase interest and motivation by making learning tasks feel more like games. Jane McGonigal, in her book *Reality is Broken* [11], expands on this idea. There can be no doubt that millions of people are currently spending large amounts of time in playing video games. McGonigal asserts that "there are 183 million active gamers" in the United States. [11, p. 3] The term *active gamer* is defined as an individual who reports that he or she plays computer or video games for thirteen hours per week on average. (Note that this is over half of the US population.) This is not just a US trend. McGonigal reports that 100 million Europeans, 300 million Chinese, 105 million in India, and large numbers in other countries are part of the online gamer community. A theme of McGonigal's book is that these trends are not merely as a result of people having fun, but that "in today's society, computer and video games are fulfilling genuine human needs that the real world is currently unable to satisfy." [11, p. 4]

Although not an authoritative source, a Wikipedia article [16] on gamification provides useful definition: "Gamification techniques strive to leverage people's natural desires for competition, achievement, status, self-expression, altruism, and closure."

McGonigal asks, "What if we started to live our lives like gamers ... and think about solving real-world problems like computer and video game theorists?" [11, p. 7] The related but more specific research question that we are investigating here is this: Can we formulate learning activities as games in ways that facilitate students' learning of software engineering concepts through increased motivation of these students?

II. Gamification and Software Engineering Education

Some educators and researchers are beginning to study games and programming. See "Gamification In Education - Learn Computer Programming With Fun" by Kumar and Khurana [10]. Gamification also can be applied to various aspects of software engineering education. For example, one more precise research question is the question of whether the use of games facilitates student learning of computer programming languages. The underlying hypothesis is that use of games tied to programming tasks may result in students spending more time on these tasks, resulting in improved learning. Anecdotally, many of us have observed the increased learning that some students experience from the use of small problems from web sites like *codingbat.com*. Solving those small programming problems with the quick feedback through

automated testing produces a game-like experience. Some programming games are beginning to appear. For example, use of the game Cargo-Bot [https://itunes.apple.com/gb/app/cargo-bot/id519690804?mt=8] is described in "Gamification of Programming" by R. White [White]. A related trend is that of "hackathons" that are blossoming around the nation. These are often intense weekends of programming in which teams of students work intensely to create apps. Students treat this competitive programming experience as a game. The tasks of programming are treated as fun. We all know that, if we assigned these same tasks as assignments in a class, some students would view these as work to be avoided or delayed.

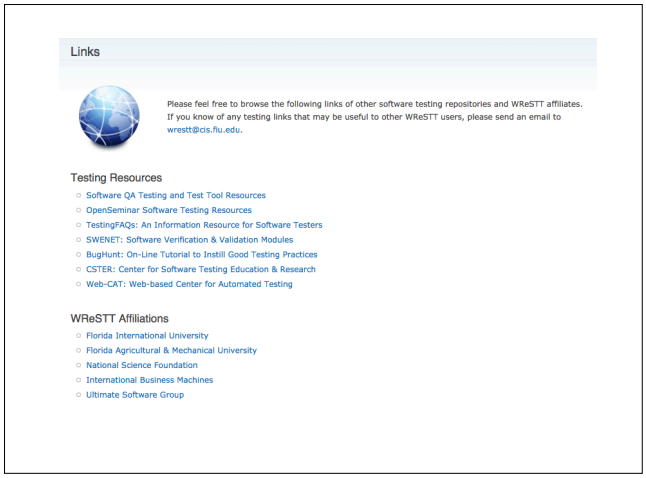


Fig. 1. Resource page in WReSTT.

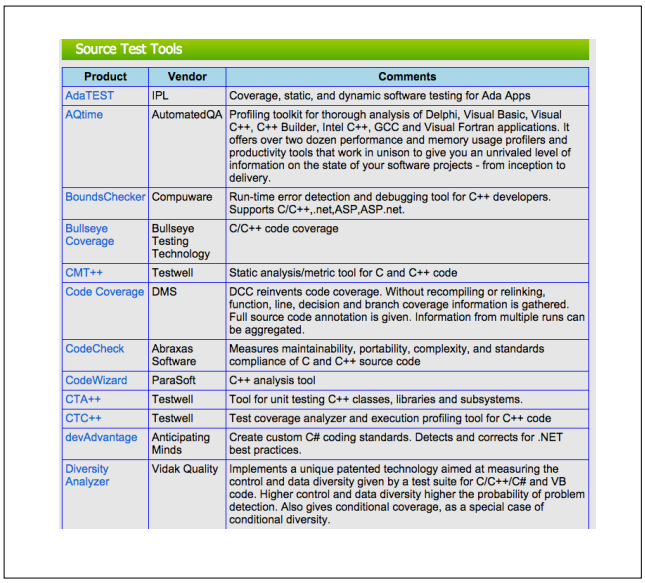
III. Teaching Software Testing

One important aspect of games is the use of rewards for success in playing games. Some types of rewards [16] include:

- Points [14]
- Badges or levels [9]
- Progress bar [12]
- Virtual currency [9]
- Competition [13]

Another more precise research question is whether games can enable greater learning in a particular subfield of software engineering. In this research project, we are studying the use of gamification in teaching of software testing concepts and testing tools. This is a question that a research collaboration, led by Dr. Peter Clarke of Florida International University, is exploring with funding from the NSF Transforming Undergraduate Education in Science, Technology, Engineering and Mathematics (TUES) program. Dr. Clarke and his students have built WReSTT-CyLE – Web-based Repository of Software Testing Tutorials: A Cyber-learning Environment web site. (See [17].) This tool is used as a test bed for this research.

This web site is a repository of information about testing and testing tools. Figure 1 shows the resource page of *WReSTT* that illustrates the links to resources about testing that are included. One very useful link is the first under the heading *Testing Resources*. Figure 2 shows a snapshot of part of the page and the types of tools to which it refers.



Source Test Tools

Product	Vendor	Comments
AdaTEST	IPL	Coverage, static, and dynamic software testing for Ada Apps
AQtime	AutomatedQA	Profiling toolkit for thorough analysis of Delphi, Visual Basic, Visual C++, C++ Builder, Intel C++, GCC and Visual Fortran applications. It offers over two dozen performance and memory usage profilers and productivity tools that work in unison to give you an unrivaled level of information on the state of your software projects - from inception to delivery.
BoundsChecker	Compuware	Run-time error detection and debugging tool for C++ developers. Supports C/C++,.net,ASP,ASP.net.
Bullseye Coverage	Bullseye Testing Technology	C/C++ code coverage
CMT++	Testwell	Static analysis/metric tool for C and C++ code
Code Coverage	DMS	DCC reinvents code coverage. Without recompiling or relinking, function, line, decision and branch coverage information is gathered. Full source code annotation is given. Information from multiple runs can be aggregated.
CodeCheck	Abraxas Software	Measures maintainability, portability, complexity, and standards compliance of C and C++ source code
CodeWizard	ParaSoft	C++ analysis tool
CTA++	Testwell	Tool for unit testing C++ classes, libraries and subsystems.
CTC++	Testwell	Test coverage analyzer and execution profiling tool for C++ code
devAdvantage	Anticipating Minds	Create custom C# coding standards. Detects and corrects for .NET best practices.
Diversity Analyzer	Vidak Quality	Implements a unique patented technology aimed at measuring the control and data diversity given by a test suite for C/C++/C# and VB code. Higher control and data diversity higher the probability of problem detection. Also gives conditional coverage, as a special case of conditional diversity.

Fig. 2. Tool References in WReSTT.

WReSTT also supports creation and deployment of learning objects about software testing and testing tools. Figure 3 is a snapshot of two of the available learning objects. These learning objects contain tutorials about a specific topic in software testing and quizzes to assess students' understanding of this information.

The WReSTT web site also provides support for creation of learning objects. In addition to text-based tutorials and associated quiz assessments, learning objects can include video tutorials and tool demonstrations.

WReSTT also includes support for social aspects of learning and game-like activities in which student teams are rewarded with "points" for spending time on learning activities and passing related quizzes. The intent and our hypothesis is that the team work and game-like nature of these activities will motivate all students to spend more time in learning these concepts and tools in support of software testing.

The intent is for WReSTT to be used as a supplement in software engineering and computer science courses, particularly those that involve programming. This includes the introductory program sequence of courses, but also more advanced courses that incorporate programming applications or projects as major learning objectives. It is this use of WReSTT as a supplement to courses that our research collaboration is studying.

Several previous studies of the use of WReSTT in courses have been reported in [2], [3], and [4]. During the 2014-15 academic year, additional studies will be completed to explore the effectiveness of gamification in teaching software testing.

IV. Future Work: Motivation of Female Students

The lack of female students in computer science is a well-known problem. Nationwide, the number of females majoring in computer science is under 15%. Some recent research evidence points to the conclusion that some females are motivate by altruistic tasks and activities. Many games are designed to appeal to the players' competitive nature. However, there is a group of games that are designed to appeal to shared goals, social interactions, and the common good. This is a direction of research that we are following here at Miami University in a scholarship project that has been funded by the S-STEM program of the US National Science Foundation. Our specific research question for this project is "Can learning games whose goal is social interaction or advancing a common good (rather than competition) be effective in attracting more females to computer science."

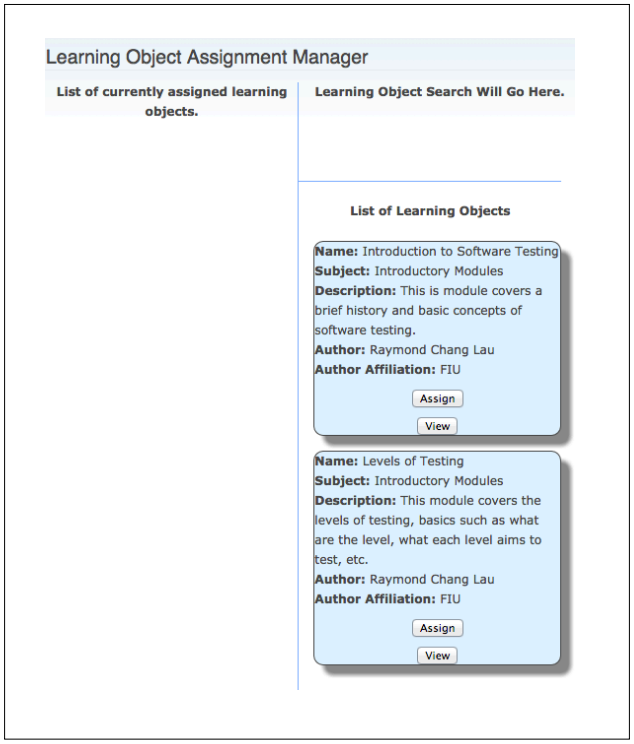


Fig. 3. Learning Objects in WReSTT.

There is a body of research that is producing evidence that motivational factors [1, 8] are a key to the gender difference in STEM fields. Since software engineering and computer science are STEM fields, we believe that this research applies here. The evidence points to the *goal congruity perspective* issue as one of these motivational factors. This *goal congruity perspective* issue is the disparity between what women want to do in their careers to what they perceive that STEM fields will allow them to accomplish. [5, 6] Women and some men seem to prefer communal goals like working with people, community service, being helpful to others; as opposed to goals that focus on winning, power, recognition, mastery; that is, more individualistic goals. [Diekman 2011] Some experiments, as in [Diekman 2011], indicate that describing the communal aspect of careers can change women's attitudes toward some STEM careers.

One hypothesis in our research investigations is that we can create digital games that can help students learn software engineering and computer science concepts, programming, etc. and do this in a way that helps emphasize the communal aspect of careers in computer science or software engineering with the outcome of an increase in women's interest in these fields.

There are many existing games that emphasize cooperation or working for the common good. *Free Rice* is a game aimed at ending world hunger. (See http://freerice.com.) The game *Groundcrew* (http://groundcrew.us) allows teams of agents to work on any social problem. *The Comfort of Strangers* is a social street game (http://swarmtoolkit.net). These are not examples of games designed the teach software engineering, but are examples of games that may attract females students and thus serve as models for software engineering-oriented games. The challenge is to create games that both teach software engineering and do this in a more cooperative, less individualistic way.

Acknowledgment

WReSTT-CyLE is a TUES II project; a collaborative effort between Florida International University, Alabama A&M University, Miami University - Ohio, and North Dakota State University. The industry partners include: Ultimate Software, IBM, and Microsoft. The first version of WReSTT, a CCLI Phase I project, was a collaborative effort between Florida International University and Florida A&M University. This material contained in WReSTT web pages is based upon work supported by the National Science Foundation under Grant Numbers (DUE-1225742, DUE-1225654, DUE-1224411, DUE-1225972, DUE-0736833, DUE-0736771).

References

[1] Ceci, S. J., Williams, W. M., & Barnett, S. M. (2009). Women's underrepresentation in science: Sociocultural and biological considerations. Psychological Bulletin, 135, 218–261. http://doi.org/10.1037/a0014412.

[2] Clarke, Peter J., Debra Davis, Tariq M. King, Jairo Pava and Edward L. Jones. ACM Transactions on Computing Education. Integrating Testing into SE Courses Supported by a Collaborative Learning Environment. In press.

[3] Clarke, Peter J., Debra L. Davis, Raymond Chang-lau, and Tariq King. In Proceedings of the 121st American Society for Engineering Education (ASEE) - Software Engineering Constituent Committee Division Track (SWECC) 2014. Observations on Student Use of Tools in an Undergraduate Testing Class. Paper id: 10123, 16 pages.

[4] Clarke, Peter J., Yujian, Fu, James Kiper, and Gursimran Walia. In Proceedings of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14). Workshop on Integrating Software Testing into Programming Courses (WISTPC 2014). March 5th, page 739.

[5] Diekman, A. B., Brown, E. R., Johnston, A. M., & Clark, E. K. (2010). Seeking congruity between goals and roles: A new look at why women opt out of science, technology, engineering, and mathematics careers. Psychological Science, 21, 1051–1057

[6] Diekman, A. B., Clark, E. K., Johnston, A. M., Brown, E. R., & Steinberg, M. (2011). Malleability in communal goals and beliefs influences attraction to STEM careers. Journal of Personality and Social Psychology, 101, 902–918.

[7] Ericsson, K. Anders, ed. The Cambridge handbook of expertise and expert performance. Cambridge University Press, 2006.

[8] Ferriman, K., Lubinski, D., & Benbow, C. P. (2009). Work preferences, life values, and personal views of top math/science graduate students and the profoundly gifted: Developmental changes and gender differences during emerging adulthood and parenthood. Journal of Personality and Social Psychology, 97(3), 517–532.

[9] Hamari, Juho; Eranti, Veikko (2011). "Framework for Designing and Evaluating Game Achievements". Proceedings of Digra 2011 Conference: Think Design Play, Hilversum, Netherlands, September: 14–17.

[10] Kumar, Balraj and Parul Khurana, "Gamification In Education - Learn Computer Programming With Fun", International Journal of Computers and Distributed Systems, Vol. No.2, Issue 1, December 2012, ISSN: 2278-5183.

[11] McGonigal, Jane (2011). Reality Is Broken: Why Games Make Us Better and How They Can Change The World. Penguin Press. p. 122. ISBN 978-1-59420-285-8.

[12] O'Brien, Chris (October 24, 2010). "Get ready for the decade of gamification". San Jose Mercury News.

[13] Reeves, Byron, J. Leighton Read (2009). Total Engagement: Using Games and Virtual Worlds to Change the Way People Work and Businesses Compete. Harvard Business Press. p. 177.

[14] Sutter, John D. (September 30, 2010). "Browse the Web, earn points and prizes". CNN.

[15] White, R. "Gamification of Programming" http://ioseducation.wordpress.com/2013/05/28/ gamification-of-programming/

[16] Wikipedia page: http://en.wikipedia.org/wiki/Gamification

[17] WReSTT: Web-based Repository of Software Testing Tools. http://wrestt.cis.fiu.edu/cen/