Serious Game Development as an Iterative User-Centered Agile Software Project

Hazeline Asuncion, David Socha, Kelvin Sung
Computing and Software Systems
University of Washington, Bothell

Scott Berfield, Wanda Gregory
Center for Serious Play
University of Washington, Bothell

ABSTRACT

Commissioned by the campus Office of Admissions, we have built a series of three campus tour and orientation games over the past academic year with undergraduate student project teams. Based on well-established game industry practices we followed an iterative agile process with Scrum and managed to avoid many classical pitfalls in game development. While we achieved some measure of success, in post-project analysis, it becomes obvious that our process would have benefited from the heavy emphasis of "users" in the User-Centered Design (UCD) methods. In this position paper, we propose that the serious game development community continue to critically analyze the results from the UCD projects to benefit from its lessons, well-understood good practices, and development paradigms.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management–*software process models*; K.8.0 [**Personal Computing**]: General–*games*; K.3.2 [**Computers and Education**]: Computer and Information Science Education–*computer science education*

General Terms

Design, Experimentation.

Keywords

User-Centered Design and Development, Serious Games.

1. INTRODUCTION

A serious game is a computer videogame designed with a primary purpose other than pure entertainment, e.g., military training or flight simulator games. When designed and implemented properly, serious games can take advantage of the engaging nature of videogames effectively and be powerful tools [12]. It is important to critically examine the development process of serious games to understand the different ways non-entertainment objectives can be explicitly integrated into videogames.

As in the case of other higher education institutions, the Office of Admissions (OOA) at the University of Washington Bothell organizes campus tours to accomplish two main goals: to introduce potential students and interested public to our beautiful campus, and to feature our talented students as the guides for the tours. The Game Across Multiple Environments (GAME) research group and the Center for Serious Play took on the challenge of accomplishing the same objectives by building customized games for the visitors. The game world and gameplay would introduce visitors to our campus, and the games would be built exclusively by our undergraduate students to feature the students' creative, artistic, and technical talents.

Over the past academic year, we have built a series of three games, one per each academic term, designed to guide potential visitors to discover the campus virtually online and to attract them to visit the campus physically on-site. In these ways, these are serious games that are standalone "*Advergames*" with in-game product (the campus) placements [14]. Our modest goals are to use simple advertisements to spark interest and elementary player self-familiarization with our campus.

With a different undergraduate team each term, we followed the iterative design (e.g., [6]) and agile development with Scrum (e.g., [11]) processes that are well-established in game industry. Based on disciplined control of feature sets and tightly following the schedule, we managed to successfully deliver all the games on schedule while avoiding classical pitfalls in game development.

However, one shortcoming we did not successfully avoid was the classical confusion between "*end users*" and "*customers*" [3]. Instead of targeting the potential visitors, our process fine-tuned the games for the OOA staff members with whom we have worked closely. The clear classifications of different user types in the agile User-Center Design (UCD) methods [3] were missing from our process. Further analysis of recent results from UCD projects reveal that these projects have many other similarities with typical game development projects. Serious games, with the typical multiple goals (i.e., fun and serious objectives), and the different types of stakeholders (e.g., end users vs. clients), could greatly benefit from the UCD methods' heavy emphasis of meeting the end-user requirements.

This paper surveys relevant results from serious games and UCD, describes our project, our development process, and the resulting serious games, evaluates our games from the various stakeholder perspectives, and briefly discusses similarities between UCD development and serious game development. The paper concludes with a proposal to further investigate combining UCD techniques with agile processes to improve future serious game development process.

2. BACKGROUND

The design and development of serious games are distinct from those of typical videogames. The gameplay design of serious

GAS'11, May 22, 2011, Waikiki, Honolulu, HI, USA
Copyright 2011 ACM 978-1-4503-0578-5/11/05 ...$10.00

games is especially challenging because, by definition, enticing gaming interactions should lead to achievements of serious objectives (e.g., military training [12], or learning [17]). This paper discusses the *software process* of designing and developing serious games and does not attempt to address the issues involved in creative gameplay design.

An agile development process using Scrum can be effective in producing regular videogames (e.g., [10]). Instead of comprehensive design documentations an agile method focuses on iterative refinement and incremental improvements to working software. This characteristic, combined with a collaborative working relationship with the customers, allows frequent user feedback that supports rapid tuning of a game to focus on promoting enticing gameplay [11]. In an education setting the fixed schedules of academic quarters and the fact that students are typically taking other classes meant it was important to adopt a variation of the sprint cycles and Daily Scrum [5, 15].

There are limited published results dedicated to the discussion of serious games development processes. Tran and Biddle [18] studied the support for collaborative design when building serious games and concluded that respect, communication, and shared conceptual model of the objective are important factors. It is important to note that there are typically multiple types of "*users*" or "*stakeholders*" in serious games development. From the "customers" who commission the game, to the "domain experts" who have in-depth knowledge of the subject area, to the "end-users" who would eventually play the game. The development process of serious games should identify and integrate the inputs from all these stakeholders.

The user-centered design process (UCD) "is an approach to interactive software system development that focuses specifically on making systems usable," [1]. There are many UCD software process models proposed, such as the Star lifecycle model [7], the Usability Engineering lifecycle [13], and the ISO 13407 Human-centered design process for interactive systems [16]. A commonality across these different models is the weaving of user involvement into the entire software development lifecycle process to ensure that the development team understands the users, the tasks that will be supported, and the context in which the system will be deployed. It is interesting that there is a lack of published results discussing approaches to combine UCD in serious game development processes.

3. OUR PROJECT

3.1 Background

In order for students to experience the entire production cycle, the development schedule for each game was limited to a 10-week academic quarter. Each game was developed by a new team of undergraduate students. These are typically graduating seniors with respectable programming skills, who are passionate about game development but who do not have much relevant experience.

While the final schedules and development team compositions of most commercial game projects are determined by the preproduction results, our end date and team talents are predetermined by the academic class schedule. For these reasons, time and technical skill constraints dictated the scope of each game in terms of features and implementation technologies. The tight constraints also meant that particular attention was paid to pre-production–which was given 1/3 of the total schedule. Careful planning allowed the team to focus on contents that would make it into the final release.

In all cases the teams took advantage of public domain software development support tools. Source code and art assets were shared via public *Assembla* subversion, and bugs and milestones were tracked via *FogBugz*. In addition, a discussion bulletin board, email, and instant text messages were used extensively.

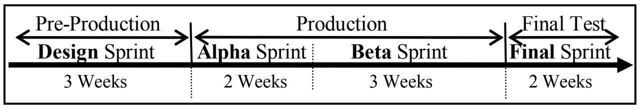

Figure 1: Project Phases and Sprints

3.2 The Development Process

As illustrated in Figure 1, all three games followed a similar process, informed by the good practices from game industry [6, 11], where the work went through three phases–pre-production (weeks 1-3), production (weeks 4-8), and final test and release (weeks 9-10). A Scrum process with variable length sprints, similar to the approach of Schild et. al. [15], was adopted in response to the tight schedule. Four sprints whose durations were driven by the amount of work involved were scheduled into the three production phases with the faculty member assuming the Product Owner and Scrum Master roles. Because the student teams had full-time academic workloads, it was difficult to use daily standups to track and refine progress. Instead, the teams met weekly for formal progress assessments and backlog refinements with frequent informal inter-member communications and small-scale task adjustments during the week.

Figure 2: Results from Brain Storming

The pre-production phase of each game started with the project team joining the campus tour and interviewing the OOA staffs and tour guides. This was followed by familiarization with existing games and finally the ideation and brainstorming aimed at extending the collective campus tour coverage of the games. Figure 2 shows the results of typical brainstorming sessions. As the ideas were whittled down to the core concept, storyboards (e.g., Figure 3) and concept write-ups were prepared.

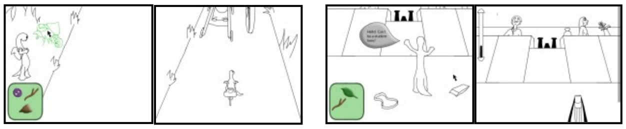

Figure 3: Examples of Storyboards

Credit: Katherine Phillips

These materials were used to pitch the game idea to the OOA staffs and tour guides and were refined based on their feedbacks. Once the main required elements were defined, a prioritized

Product Backlog was prepared and tasks were pulled by team members with occasional guidance by faculty member to balance team workload. At the final stage of pre-production, sample game scenes were sketched (e.g., Figure 4), with simple prototypes were built, and game engines were evaluated for final decisions on the required feature set.

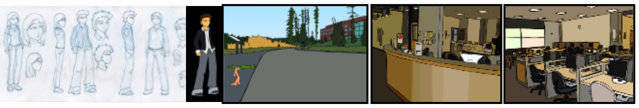

Figure 4: Examples of Pre-Production Style Sketches
Credit: Aaron Amlag

During the production phase, formal playtest sessions were instituted at the end of Alpha and Beta Sprints where the development teams would observe the OOA staff playing the games, take notes, and receive formal written evaluations.

Then Final Sprint was dedicated to final fine tuning and bug fixing. One last formal playtesting was scheduled before the final release where all students were invited to participate. Based on these results the game was adjusted, where possible. At the end of this time, the product was formally released.

4. PROJECT RESULTS

Following the above process, a series of three games have been built in consecutive academic terms: a web-based click-to-explore game to introduce the campus; a Zune-based augmented reality game for players to solve a mystery while physically touring the campus library; and an Android-based augmented reality game that guides new students to physically visit various campus locations (e.g., the cashier, security office).

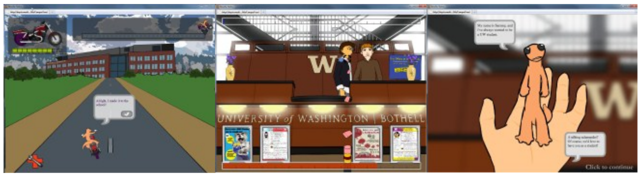

Figure 5: Sammy UWB – Web Game
Credit: Aaron Amlag (lead artist), Bryan McMahon, and Katherine Phillips

Figure 5 shows the screen shots from the web-based on-line game.[1] The entire game is carried out in our campus environment. The right-most screen shows that at the end of the game, the player is invited to visit the campus physically to continue with the campus tour game.

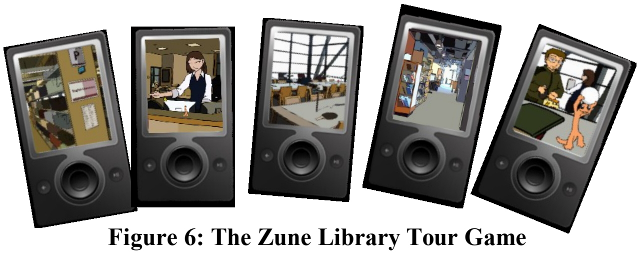

Figure 6: The Zune Library Tour Game
Credit: Aaron Amlag (lead artist), Shane Krolikowski, and Scott McPherson

Figure 6 shows images from the Zune library tour game. Players are challenged to explore the library following clues to solve a "*who did it*" mystery. This game takes advantage of the Zune devices' WiFi proximity detection mechanism to integrate physical location into the gameplay.

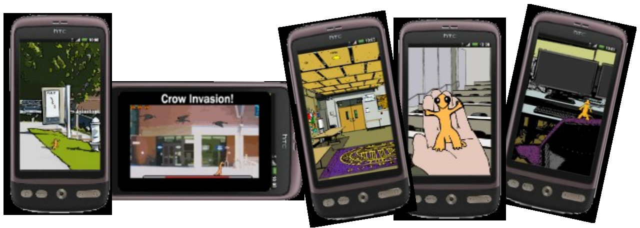

Figure 7: The First Day Registration Game
Credit: Ryan Hoaglan (*dual role as artist*), Sidney Maxwell, Dmitry Ryzhkov, Kimberly Walker (*dual role as artist*)

Figure 7 shows the Android phone registration and orientation game. This game uses the GPS for outdoor and Bluetooth sensor for indoor location tracking. Similar to the Zune library tour, the physical positions of the player are used as mechanics for advancing the gameplay.

Each game was released immediately at the end of the corresponding academic term. The end of each game includes links to online surveys for soliciting user feedback. It is interesting that while the games received overwhelmingly positive feedbacks from the University staffs, especially from those of the OOA, the number of online surveys received were small. Informal feedback from students verified that while most felt that the games reflected positively upon the campus, there seemed to be a general lack of interests in playing the games.

5. EVALUATIONS

Informal retrospectives with the development teams revealed that students had an overall positive experience in developing the games. We were pleasantly surprised by their abilities to quickly learn new tools (e.g., Actionscript, or programming with sensor devices) and to deliver quality products, despite a tight schedule and lack of previous experience.

After the academic year, we had the opportunity to reflect and critically analyze our process. The teams' ability to deliver the products can be attributed to following the industry practice of dedicating a significant percentage of the schedule to pre-production and adopting game feature sets to the tight schedule and team talents. It is reassuring that the teams have managed to avoid some of the challenges in *typical* game development projects (e.g., unachievable ship date, feature creep) [11].

We received different feedback from our customers and our end-users. Our customers, the OOA and other University staff, loved the fun mini-games and the informative coverage of the campus. For example, the librarians are especially pleased with the thorough library coverage of the Zune game. Our games, however, were not well-received by our end-users who were the incoming students. A typical response from the students was "*why am I playing this game*?"

A close analysis of the lack of student interests in the games reveals that our process has failed to differentiate between the "*end-users*" from the "*customers*" [3]. The process we followed

[1] This game is accessible via http://depts.washington.edu/itls/CampusTour

led us to design and fine-tune the games to meet the customer expectation instead of the end-user's perspective. In our case, important factors such as community and social interactions that were not explicit goals of our customer but are important to our end-users, were missing from our games.

Partially due to the lack of experience, we did not clearly differentiate between our customers (OOA staffs), domain experts (tour guides), and end-users (new students). This is a lesson learned and we should modify our development process accordingly. Our mistake might have been avoidable if we had used an elaborate and explicitly user-centered method (e.g., [3]).

6. UCD AND SERIOUS GAMES

UCD principles clearly identify the end users as the core of a software project [1]. A recently described user-centered agile method further differentiates between *direct users*, *indirect users*, *customers*, *stakeholders* and detailed a process that integrates the inputs from all these users [3]. Our development process would have benefited from such clear distinctions of user types.

Other techniques used in UCD include contextual inquiry (observation of people in the context of their work), interviews, paper prototypes (screen layouts on paper), personas (characteristics of user roles based on customer data gathered), and prototype usability tests (users interact with a running system) [2, 8, 16, 19]. The similarities to the typical techniques employed in game development processes (e.g., [6]) are striking. Yet, a literature search reveals that there are currently little or no cross pollinations between UCD and game development.

Many UCD projects contain similar challenges and uncertainties as serious games (e.g., the early attempts at building multimedia applications on mobile devices [9] or building an ATM interface for illiterate users [4]). These applications must guide and support their users to accomplish specific tasks through graphical objects and interfaces. Though not exactly identical, both serious games and these examples involve non-conventional challenges that require creative solutions. In addition, the graphical-oriented interaction and the task-oriented nature of the applications, suggest potentially interesting overlaps in the design and development process between the two communities.

Finally, as illustrated by our experience, taking a UCD approach to development can be especially important and enlightening for serious game development where the customers are typically not the end-users and the game designers/developers usually do not fully understand the problem domains. Thus, we plan to further investigate how UCD techniques, combined with agile processes, can guide future game development and can enhance the end-user play experience.

ACKNOWLEDGMENTS

Xheni Diko, Hang Dang, Jill Orcutt, Lindsey Wille, Sabrina Moss, and the entire team at the OOA offered their tireless support throughout the project. This work was supported in part by Microsoft Research under the Computer Gaming Curriculum in CS RFP, award numbers 15871 and 16531, and a grant from the Teaching & Scholarship Enhancement Project, University of Washington, Bothell (UWB). All opinions, findings, conclusions, and recommendations in this work are those of the authors and do not necessarily reflect the views of Microsoft, or UWB.

REFERENCES

[1] Web accessibility initiative. Notes on user centered design process, http://www.w3.org/wai/redesign/ucd.

[2] B. Broschinsky, Using persona with XP at LANDesk software. *Proc. of the AGILE Conference*, 2008.

[3] L. Constantine. Process agility and software usability: Toward lightweight usage-centered design. *Information Age*, Aug/Sep 2002.

[4] A. Cremers, et. al. User-centered design with illiterate persons: The case of the ATM user interface. In K. Miesenberger, et. al., *Computers Helping People with Special Needs*, Vol. 5105 of *Lecture Notes in Computer Science*, PP. 713–720, 2008.

[5] C. Fernández-Vara and P. Tan. The game studies practicum: applying situated learning to teach professional practices. *Future Play'08*, PP. 25–32, 2008.

[6] T. Fullerton. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. Morgan Kaufmann, second edition, 2008.

[7] H. R. Hartson and D. Hix. Toward empirically derived methodologies and tools for human-computer interface development. *International Journal of Man-Machine Studies*, 31:477–494, 1989.

[8] K. Holtzblatt, B. Wendell, and S. Wood. *Rapid contextual design: a how-to guide to key techniques for User-Centered Design*. Morgan Kaufmann, 2005.

[9] Z. Hussain, et. al. User interface design for a mobile multimedia application: An iterative approach. In *Adv in Computer-Human Interaction,* PP. 189 –194, 2008.

[10] C. Keith. Scrum rising. Game development magazine, http://www.clintonkeith.com/resources/scrumrising.pdf, 2007. Last access: 01.13.2011.

[11] C. Keith. *Agile Game Development With Scrum*. Addison Wesley, 2010.

[12] H. Kelly, K. Howell, E. Glinert, L. Holding, C. Swain, A. Burrowbridge, and M. Roper. How to build serious games. *CACM*, 50:44–49, July 2007.

[13] D. J. Mayhew. *The Usability Engineering Lifecycle*. Morgan Kaufmann, 1999.

[14] B. Sawyer and P. Smith. Serious games taxonomy. *http://www.seriousgames.org/presentations/serious-games-taxonomy-2008_web.pdf*, 2008. Last Access: 1/27/2011.

[15] J. Schild, R. Walter, and M. Masuch. Abc-sprints: adapting scrum to academic game development courses. *FDG '10*, PP. 187–194, 2010.

[16] H. Sharp, Y. Rogers, and J. Preece. *Interaction Design: Beyond Human-Computer Interaction*. Wiley & Sons, 2007.

[17] K. Sung. Computer games and traditional computer science courses. *CACM*, 52(12):74–78, December 2009.

[18] M. Q. Tran and R. Biddle. Collaboration in serious game development: a case study. *Future Play*, PP. 49–56, 2008.

[19] J. Ungar. The design studio: Interface design for agile teams. *Proc. of the AGILE Conf.*, PP. 519 – 524, 2008.