

Games for Teaching Software Development

Bruce Scharlau
Computing Science
University of Aberdeen
Aberdeen, AB24 3UE, UK
+44 (0)1224 272193
b.scharlau@abdn.ac.uk

ABSTRACT

We have been using games to teach software development for a number of years now with our MSc IT students as a warm-up to the group project over the summer. Games, and game-like processes without accompanying lectures can provide the participants with enough insight into the thinking and rationale of agile, lean, and kanban approaches to software development. This weeklong crash course consists of games and exercises to provide students with a good foundation in software development processes. The agile games, and game-like service design activities along with some Lego® Serious Play® games, illustrate why we use the agile approach. The game environment provides a mixture of icebreaker, team building and software development aspects in a useful week of work that the students enjoy.

Categories and Subject Descriptors

K.6.3 [Computing Milieux]: Software Management – *software development*

General Terms

Management, Human Factors

Keywords

Agile, lean, games, software development, Lego Serious Play, service design

1. INTRODUCTION

We use games to teach agile practices and concepts to our students. We use the same games agile coaches use to train their software development teams. While agile games have been developed for teaching students agile software development using card game simulations [4], we have used simple games to illustrate specific aspects of software development. We have also started to use StrategicPlay® processes with Lego® Serious Play® to cover team identity, and shared conceptual aspects too [10].

Play is an important aspect of life, which many tend to reduce as they age. We forget about its learning aspects of play [2]. We might assume students will ‘play with the code’ in lab exercises, and usually leave it at that.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITiCSE’13, July 1–3, 2013, Canterbury, England, UK.

Copyright © ACM 978-1-4503-2078-8/13/07...\$15.00.

Play, and games in particular, however, provide a number of important features to aid learning. First, play provides a safe environment where failure is acceptable as it ‘is only a game’ [7]. Second, as play is voluntary, we are not coercing anyone to participate. Third, play can provide a situation where we can lose ourselves in the activity, as time seems to fly past. Fourth, play is fun, and something, which we want to continue as and when we can. Lastly, play is improvisational. [2] Together the attributes of play create a situation conducive to learning for our students.

Learning games used in this play setting should follow the ‘SCARF’ model and address: status, certainty, autonomy, relatedness and fairness. The players should understand their status in the game, while the rules provide certainty of outcome, and the players have autonomy within the game scenario. In addition, there should be relatedness between the players so that they are seen as friendly players, while the rules provide a fair game to all participants. [17]

When all of this comes together you find your players are in ‘flow’. They are sufficiently challenged to keep their interest in the game, while not being overly stretched by frustrated gameplay. This is the same experience you find with gamers, and developers who lose hours of time when in ‘flow’. When in flow they are learning the most so the aim is to have them achieve this state as quick as possible, and to engineer conditions to keep them there as long as possible while playing the game. [18]

Games can provide an engaging way to learn. Participants are caught up in the excitement, and learn more deeply than they would by listening to a lecture, or reading about the topic. In particular games, which draw on physical activity along with thinking processes, deliver the fastest learning as they increase the blood flow and therefore oxygen supply to the brain so participants are ‘smarter’ compared to sitting still. [11] In addition, when you can have creative games where participants are thinking through their hands, such as with Lego Serious Play [10], then you are also tapping into implicit knowledge, which is otherwise hard to access. [18]

As described previously [19], we introduced agile games to the beginning of the group project for our MSc Information Technology students several years ago. The rationale for this was that we wanted to ensure all of the students had a good understanding of the agile development process before starting the summer MSc projects.

A number of agile games found at TastyCupcakes.org, a site fuelling ‘innovation and learning’ [24] introduce the issues of client participation, estimating and planning iterations [3], lean and kanban work limiting approaches [14, 1]. One game explores the business value of components to be built on a project. Another game looks at the whole Scrum process, while a third compares the value of limiting work in progress by using kanban principles.

We also use a StrategicPlay team building game based on a Lego Serious Play process in order to validate our team membership

decisions. The game was designed to express individual identity and to help develop a team identity.

We use each game to introduce and explore a specific topic in each instance. When we started using games we used to deliver a lecture to introduce the topic prior to running the game. However, we have since thrown out the lecture and let the game carry the whole effort. Yes, we talk about why we were using the game and what points the students should pay attention to while playing the game in order to set the context. We also run a retrospective debrief after each game to draw out the lessons learned and put them into the context of what the students would be doing in the coming weeks. This combination worked well: briefing, game, debriefing.

Each game relates to one or more processes each team will use in the coming weeks during their group project. Each student hears about the topic, simulates using the tools and processes in the game, and then uses the real tools and processes during their project. The sessions provide a slight variation on ‘see one, do one, teach one’ so students can help each other in the practical sessions and over the summer.

2. WE ONLY RUN AGILE PROJECTS

All of our summer group projects comprise the MSc component of the degree for the one year long MSc Information Technology degree. All of the student teams have live clients, who are either from external organisations, or are units within the university who want an application developed to help with their work. The twenty or so students on this programme each year all did something other than computing for their first degree, and are now learning the basics of computing and software development. Most continue afterwards as business analysts or developers, while a small number continue onto a PhD programme.

We have organised live clients for the MSc summer projects for four years now and have found it is better for the students than if computing staff mentor ‘pretend’ projects. Real clients provide more motivation and a better experience for the students than departmental staff. Live clients also end with something they can use after the summer too.

The students all use agile project processes for the projects in order to set up a double-loop learning [21] system so that both the client and the students learn more about the product being developed. This obviates the need for big up front design, and encourages activities to aid learning about the users and the product itself by starting with a minimal viable product pushed out as soon as possible in order to learn more about the product and its users. [16]

An agile development process should ensure that there is something usable for the client at the end of the project. We also want to know that students have a taste of what a larger commercial agile project might be like. The course on software development, which they take as part of the programme, covers traditional project processes as its coursework assessment, as this is what is often used by many organisations. However, students should also experience agile projects, so that they know the difference, because those who end up working with smaller software houses will usually work in an agile way too.

Learning agile processes in a safe game setting, and to then use them in their own MSc project work, helps to embed the

processes in the students’ understanding. Teaching the materials without this close connection between concept and application does not work as effectively. When tried with regular assessment work, the students are usually reluctant to pursue agile practices beyond the ‘focus on one story at a time’ aspect. We find too that students usually use test or behavior driven development (TDD/BDD) for coursework only when it is required for the submission of their Ruby and Rails assessments. Students rarely do this voluntarily [8]. When shifted to a larger project with a live client the case for tests becomes more convincing and they make the connection more easily between their own work and testing, for example, without prompting.

3. THE PROJECT SETTING

Using live clients provides a demand for working projects at the end of the summer. Previous clients have included international and national charity organizations, community groups, and small businesses, along with units within the university. We have not had a problem finding clients for the teams since we started using real clients four years ago.

Using real clients on a twelve week project means the students need to shorten the time between when the projects start and they can deliver a minimal viable product to the client with which to aid the learning about the users and the product being developed. Agile processes along with appropriate business model [13] and service design approaches [23] help to deliver this quickly.

Using agile games means they can experience the use of agile processes before they use the processes with real clients. This helps them understand why they are doing what they do when they start work on the project.

The week prior to the project starting has games to bring all of the students up to a common starting point. The games cover agile development and basic aspects of service design, and we provide the teams with a mini-project scenario to practice against. We want to ensure that those who started in January are close in agile software development understanding to those who started in September. Teams comprise a mixture of both student cohorts to aid this transference of understanding between the students.

4. THE GAMES

The agile games themselves are simple in concept and design. Each game has a number of iterations played with variations each session to help inform the players of the game’s message. While there are a number of agile games available at TastyCupcakes.org and elsewhere as noted in the references, we have focused on ones emphasizing aspects that highlight the importance of building the correct component at the right time according to agile and lean principles.

As the author is a trained StrategicPlay facilitator in the Lego Serious Play process, we also use a StrategicPlay identity session using a Lego Serious Play process as the first game for the students in order to (a) help build each team through a shared experience, and (b) help them get to know each other better. The session runs as follows: each student models metaphors in Lego of what they can bring to the team, and to the project itself. After building the model each person then explains the model to the others in the team, and anyone can ask questions about the model. The questions in the session are always about the model, which

becomes a source of shared understanding. It shows that each member of the team has something of value to bring to the team.

The second game we use is based on an empathy map [7] to develop different user personas for the applications. This is best done on a wall with sticky notes and a poster showing quadrants for what the persona ‘think and feels’, ‘sees’, ‘says and does’ and ‘hears’. These can be refined and incorporated into the mini-project as practice for processes to use with their real project.

A new game based on prototyping [15] the mini-project applications is also played. We use paper prototypes of interfaces to trial the application concept with other people to see whether what the developers conceived does indeed match what the users need. We have used paper ones, but other options such as role-playing can also be done to illustrate a variety of approaches. The goal of this activity is to better understand user needs and to explore alternative possibilities to meet those needs in the best way possible.

The fourth game we use is the Lego4Scrum [20] simulation, which sees the participants building a city from Lego using the scrum process of sprints based on the prioritised backlog of the product owner. This game divides the class into teams, who together build one city. Each iteration comprises an estimate, plan, build and retrospective phase. At the end of each iteration the product owner checks whether what was built matches the user story and then ‘signs off’, or rejects each feature. Only completed features count in the iteration, and rejected ones are finished in the next iteration. The Lego4Scrum game offers possibilities for students to understand of the whole software development process. In particular, students can see how different parts developed by different teams, or individuals, contribute to the whole application.

The fifth game the students play is the Kanban Game [9]. The purpose of the game is to experience what happens to the workflow when limits are placed on the number of items that can be worked on each stage of the project. The game starts with no work in progress limits to any of the columns (Do, Doing, Done) and then adds limits in subsequent rounds of the game. Counter intuitively, the amount of work that can move through to the ‘done’ column speeds up as they shift from each person working on one item across the board to one where each focuses on specific tasks with limits placed on the numbers of items in the ‘doing’ column and each player ‘pulls’ work to them as they have space available.

The sixth game we use is the Offing the Off-Site Customer Game [12]. The purpose of this game is to illustrate that it is better to have developers liaise directly with clients than it is to work through business analysts who translate communications with the client for the developers. It also shows that tests need to be developed with the client’s direct input in order to validate the features being developed so that they can be used as a proxy for the client when developing the application.

We have also started trialling StrategicPlay shared model building and scenario testing sessions using Lego Serious Play processes when teams start their project with their client. This is used to ensure that everyone had common shared concepts in mind when working on the project over the summer. It has also meant the students and the client had more time to become familiar with one another.

5. IMPLEMENTING THE GAMES

Each game is situated in the context of the mini-project being run that week as well as the group project for the summer. The students play the games with the same teams they will use for the summer projects, in order to foster team building, and can make the connection from playing the game to what they need to do as a team for the project.

The StrategicPlay identity session using a Lego Serious Play process follows a routine of model building phases followed by storytelling phases as each person explains what the model of the metaphor they built represents. This ensures that everyone speaks and listens to what the others are saying too. The focus of each round is determined by the facilitator’s questions. This is explained more fully in the Lego Serious Play guide [10].

After general skills building of metaphors with Lego the session has participants showing what they bring to the team, and then asks members to add something to another person’s model that they think is missing. Lastly, each person models an aspect of team life, which could be a fear they have of teamwork, or something they enjoy about teams. This helped identify issues teams would need to address in the coming weeks and helped uncover elephants in the room.

The identity session helps highlight potentially dysfunctional teams, which could be broken up and its members distributed among the other teams. This is good to discover early in the project phase, and is an issue will not usually show up in any of other activities we run that week.

The Lego4Scrum game comprises multiple rounds of students using Lego bricks to build as many of the requested features as possible in the game iterations. The accompanying notes for the game make implementation easy, and the hard part is forcing the students to estimate the features, plan which ones can be built in the time available, and then build them. To counter the urge to play with the Lego before the ‘build’ phase, staff need to decompose any ‘prototypes’ built before they finish the estimating and planning phase. During the product owner review phase product owner needs to be sensible about what is ‘complete’, but also point out in the retrospective phase what can be done differently to avoid problems.

The estimating phase of the Lego4Scrum game is possibly too easy. Maybe this should be made harder by shortening the build time. We have run the game a few times now with different audiences and seven minutes means many things can be built in that time. Either more tasks (buildings) need to be assigned to each team for each iteration, or as suggested the iterations should be shorter. The teams start from rough estimates and move forward based on the experience they gain together. As they move through the iterations the three stages run more smoothly.

An interesting thing about this game is that only after buildings are rejected do players think to ask the product owner questions about what will be accepted for a building. This has proven true when the game has been facilitated at different sessions with student as well as professional developer participants. Once this hurdle is achieved, then the town grows much faster as communication between the teams and the product owner grows.

The Kanban game comprises three rounds of the students moving features across a kanban board. Each player seeks to move the features in his or her care to the ‘done’ column. The game runs for

three iterations using different types of kanban boards each round. The first round uses a three-column board with 'to do', 'in progress' and 'done'. The second one introduces stages for the work in progress, so it is broken down to 'design', 'develop' and 'test'. The last version adds queues to each stage of work, plus work limits equal to the number of people working on that stage.

The mechanics of the game are straightforward, but the concepts are counter-intuitive. Only by playing the game is it clear to the students why the board changes make the features move faster across the board, and that they too must reflect on their work in progress during their project. As with the Lego4Scrum game, a facilitator needs to wander amongst the teams to clarify the gameplay.

The Offing the Off-Site Customer game has several rounds to illustrate why the client needs to be closely involved with the developers. In the first round of the game a 'business analyst' writes out 'requirements (instructions)' for the developer to follow that are based on abstract drawings. During this iteration the analyst is at one table, while the developer, who recreates the drawing based on the written instructions, is at a different table.

For the second iteration the analyst sits next to the developer and tells her what to draw without showing the developer the drawing. During this phase the drawings come closer to the originals. Eventually, all students understand the lesson of the Offing the Off-Site Customer Game, that clients need to speak to developers to derive sensible requirements with clear feedback from the client about whether the application is being developed 'correctly' according to the needs of the client.

The two service design games are useful. While the empathy map and prototypes were not 'games' in the sense that they did produce artefacts, which are used in the project, they were developed through a game process that moved towards fuzzy goals. These two games are 'played' where the SCARF model holds true so what happens in them is 'not real' until such time as the session is 'done' and the artefacts are accepted and used in the project. Until then the artefacts could be discarded and started over until the team is happy with their result. These are 'games' in the sense that the artefacts are created through play, and 'it is only a game' with special rules applying. The service design games also help highlight the necessity to students for early feedback on conceptual ideas before any software is built.

The StrategicPlay shared model session with the students, the client and key invited individuals, follow a regular Lego Serious Play session format with skills building and then the building of individual models by participants of their metaphors based around a number of open ended questions aimed at developing ideas around the purpose and goals of the application. For example, an open ended question might be 'what were the biggest challenges that we had to face?' With individual models completed, the participants use them to build a shared model incorporating the best of all individual metaphor models, plus anything else, which is deemed 'missing' from the combined effort. The role of models as repositories for shared concepts becomes more apparent. The storytelling aspects of the process mean that everyone knows, for example, the skeleton mini-figs represent the dead hand of bureaucracy to this group at this time.

The clients have been pleased with the result and with how quickly the StrategicPlay sessions resulted in new ideas for their applications. While intensive, these sessions are useful for all

participants as everyone leaves with a clear understanding of the project and its goal.

All of these games engender laughter and loud talking, while also providing a change of pace for students. They also show students that they can step away from their laptops, and illustrate why they should follow suggested agile, lean and service design approaches. Games can provide the 'takeaway messages' as a reliable method to ensure that students carry on the practices in their groups when they are working on their own. Lectures are not always needed. Structured game sessions can work better in some situations.

6. LESSONS LEARNED

Games and exercises put into context with briefing and debriefing sessions work instead of lectures. You do need to structure and time box the games, and game-like activities with care, but a course of games for the students is possible, and helps students learn by 'doing not talking'.

Running games in class is fun. It breaks up the routine and lets in spontaneity, as you are not sure what will happen. Most of the time it works, but sometimes it does not and you learn from that too, so even failure is a 'learning opportunity'.

Running games can be time consuming. You need to get materials together and set everything set up so that the game runs smoothly on the day. You should have a helper if possible to answer questions and distribute the various bits of kit that might be needed, and to generally help keep an eye on things.

While games have an order, and suggested timings, be ready to adapt these to your circumstances. Sometimes games take longer than planned the first time you run them, or some students are restless while waiting for others, so be ready to adapt the game and shorten a section so that the game can move on and everyone participates again. Alternatively, have other activities available for those waiting, or devise a variation of the game for next time to remove this issue.

You need to always run a debriefing session after the game. Even if the debrief is only five or ten minutes long, it helps embed the lessons the students should have learned from the session. This will help to reinforce what was said prior to the game.

In the debrief session you can go over both general and specific issues. For example, under general issues we ask: what went well, what was hard, what takeaways people will remember and what could be done better. Under specific things we ask issues relating to the key concept of the game, such as 'get early feedback', 'ask the product owner', 'validate your assumptions', 'be sure to use tests' and others.

Write down your own reflections as soon as possible after the game (as well as general notes during the game) so that you know what to change for next time. For example, notes written after running the Lego4Scrum game, enabled the game to be run again more effectively at an unconference after enough Lego was borrowed from another event participant.

Run trial sessions ahead of time if possible to avoid surprises on the day. This is especially true with games you developed. At least run the game on your own with you being all players as required. This will let you sort out any potential issues and revise the game before running it with the students.

The StrategicPlay sessions work well and we will always include an identity session for all teams in the future. The storytelling aspect of the sessions means we uncover issues not raised by other activities. Our restriction here is that these need to be facilitated by someone trained in the process, so we can only run these sessions serially at present.

7. HAVE GAMES HELPED?

There still remains the question about whether using games plus discussion has helped the learning process for students compared to the lectures about running software projects, which we used previously. This question can be answered on a number of levels.

First, student engagement with their projects went up since we started using live clients with the students. The students now worry more if there is not regular contact with their clients, as they understand the need for regular communication more so than when staff played the client role.

Second, student questions during the project have changed. Instead of questions about the management of the project over the summer, and technical questions, we now have questions about setting up a kanban board to keep track of tasks, and about different types of test frameworks for Ruby. We have found this means that the students understand the concepts of agile development and using them in their projects.

Third, the quality of the software projects submitted at the end of the summer has improved since the change. Even if we discount the change at about the same time from using Java to using Ruby as the programming language, then the overall scope and complexity of the project work has increased. The robustness of the software is also higher now that more agile practices are incorporated than previously. The teams now also deliver working software, which didn't always happen previously.

Fourth, the students seem more confident in their own abilities than previously. They can more easily understand how they will develop the application they have been given, and readily accept the challenge. In the past, there was always concern on their part that they were unable to achieve the project goals. This is no longer the case. Now they can see how to decompose the project to smaller deliverable parts that will build into a whole, which was something they had difficulty doing previously.

Introducing games in order to for students to more quickly and deeply learn to use agile approaches, appears to have worked compared to when we only used lectures plus practical sessions on software practices such as TDD and version control. The projects are of better quality, the students are more engaged, and confident in their abilities.

8. CONCLUSION

Using games in the classroom is open to everyone. There are a growing number of agile games about [24], and a growing number aimed at businesses in general, which can be used too. In addition, there is a growing list of game-like tools covering service design aspects so that the whole software development cycle through 'discovery, design, develop and delivery' is covered [5, 6, 22, 25]. Find suitable ones for you to spread the important lessons that agile can bring to software development. Some involve more preparation than others, so start looking at the games early so that you can arrange the requisite materials ahead of time.

As with all things agile, we want to 'fail fast' and provide 'learning opportunities' to learn from failures, so if one game does not work for you, then change it, or try a different one, and try again. In any case, do try games in the classroom. They are fun and add a dimension of 'learning by doing', which the students appreciate to bring another avenue of learning to the classroom.

9. ACKNOWLEDGMENTS

My thanks to the University of Aberdeen for an LTEP grant to aid the purchase of Lego kits, and to Katrin Elster at StrategicPlay <http://www.strategicplay.de/en/> for Lego Serious Play facilitation training and introducing the many possibilities of serious and strategic play.

10. REFERENCES

- [1] Anderson. D. 2010. *Kanban*. Blue Hole Press, Seattle, WA.
- [2] Brown, Stuart. 2009. *Play: How it Shapes the Brain, Opens the Imagination, and Invigorates the Soul*. Avery, New York, NY.
- [3] Cohn, M. 2005. *Agile Estimating and Planning*. Prentice-Hall.
- [4] Fernandes, J. M., Sousa, S.M. 2010. PlayScrum - A Card Game to Learn the Scrum Agile Method, in *vs-games*, pp.52-59, 2010 *Second International Conference on Games and Virtual Worlds for Serious Applications* (Braga, Portugal, 25-26 March). DOI = <http://doi.ieeecomputersociety.org/10.1109/VSGAMES.2010.24>
- [5] Festival Design DNA for service design process and tools <http://design.festivalslab.com/portfolio/download-all-materials/> accessed 9 January 2013.
- [6] Frog Design Collective Action Toolkit <http://www.frogdesign.com/collective-action-toolkit>
- [7] Gray, D., Brown, S., Macanuso, J. 2010. *Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers*. O'Reilly, Sebastopol, CA.
- [8] Isomöttönen, V., Lappalainen, V. 2012. CSI with games and an emphasis on TDD and unit testing: piling a trend upon a trend. *ACM Inroads* 3, 3 (September 2012), 62-68. DOI=10.1145/2339055.2339073 <http://doi.acm.org/10.1145/2339055.2339073>
- [9] Kanban Game <http://finance.groups.yahoo.com/group/kanbandev/message/9216> accessed 9 January 2013.
- [10] Lego Serious Play. *Open-source/ introduction to Lego® Serious Play®* https://dl.dropbox.com/u/5032997/LEGO%20Serious%20Play%20OS/LEGO%C2%AE_SERIOUS_PLAY_OpenSource.pdf accessed 9 January 2013
- [11] Medina. J. 2008. *Brain Rules*. Pear Press, Seattle, WA.
- [12] Offing the Off-Site Customer Game <http://jamesshore.com/Presentations/OffingTheOffsiteCustomer.html> accessed 9 January 2013.
- [13] Osterwalder, A. 2010. *Business Model Generation*. John Wiley and Sons, Inc., Hoboken, NJ.

- [14] Poppendieck, M & T. 2009. *Leading Lean Software Development: Results are Not the Point*. Addison Wesley, Upper Saddle River, NJ.
- [15] Ratcliffe, L., McNeil, M. 2012. *Agile Experience Design: A Digital Designer's Guide to Agile, Lean and Continuous*. New Riders, Berkeley, CA.
- [16] Ries, E. 2011. *The Lean Startup: How Today's Entrepreneurs use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, New York, NY.
- [17] Rock, David. "SCARF: a brain-based model for collaborating with and influencing others" from http://www.your-brain-at-work.com/files/NLJ_SCARFUS.pdf accessed 9 January 2013.
- [18] Roos, J. 2006. *Thinking from Within: A Hands-on Strategy Practice*. Houndmills, Basingstoke, UK.
- [19] Scharlau, B. Agile Games for Software Development in 12th Annual Conference, the Higher Education Academy Subject Centre for Information and Computer Sciences. (23-25 August 2011, Belfast, UK) http://www.ics.heacademy.ac.uk/events/download.php?file=events/presentations/257_944_ics-belfast-2011-scharlau.pdf accessed 9 January 2013.
- [20] Scrum Simulation with LEGO <http://www.lego4scrum.com/> accessed 9 January 2013.
- [21] Senge Peter. 1993. *The Fifth Discipline: The Art and Practice of the Learning Organization*. Random House Business, New York, NY.
- [22] Service Design Tools. Design activity: testing and prototyping <http://www.servicedesigntools.org/taxonomy/term/3> accessed 9 January 2013.
- [23] Strickdorn, M., Schneider, J. 2011. *This is Service Design Thinking*. John Wiley and Sons, Inc., Hoboken, NJ.
- [24] TastyCupcakes.org <http://tastycupcakes.org/> has many agile specific games, accessed 9 January 2013.