Methods and Processes Definitions for Multiplatform Social Network Games Development with Distributed Teams

Angela Peres[1] Fernando Selleri[1] Jamilson B. Antunes[1]
Fernanda Martins[1] Kellyton Brito[1,2] Rafael R. Wanderley[1]
Felipe Furtado[1] Vinicius Garcia[1] Silvio Meira[1]

[1]Federal University of Pernambuco (UFPE), Informatics Center (CIn), Brazil
[2]Federal Rural University of Pernambuco (UFRPE), Statistic and Informatics Department (DEINFO) Brazil

Abstract

In recent decades, agile methodologies have contributed in the process of developing software as an efficient way to manage requirements. Game development for geographically distributed teams includes additional elements that also need to be managed properly. This work is an experience report of a game development by geographically distributed team, using Scrum methodology. The game runs on web and mobile platforms and has interface with social networks such as Facebook and Twitter. The purpose of this article is to contribute to the discussion about Scrum's adaptation for game development with respect to the methods, process definition and tools used in conception, design, definition, prioritization of backlog items, monitoring requirements and quality assurance.

Keywords: agile methodology, social games, software development process

Authors' contact:
{alp5, fss4, jba, fnmm, ksb, rrw, fsfs, vcg, srlm}@cin.ufpe.br

1. Introduction

The games' development increased its importance in software industry and market from desktop to mobile devices [Keith 2010]. This incorporates a series of new issues and challenges in software development, because, in addition to the technological and functional requirements, includes other equally important elements for the success of the project, such as the process of intense creation occurring in parallel on the development, the design and refinement of animations, sounds and scenarios [Keith 2010].

On the other hand, in recent decades, geographically distributed teams have been adopted by software organization motivated by well succeed experiences in complex project such as Linux and Open Office [Martin and Hoffman 2007].

Agile methodologies have allowed better management of product requirements and the development process that must be adapted to this context. Which aspects of agile methodologies should be adapted in the design and development of games? What challenges arise in the context of geographically distributed teams? From the understanding of these challenges, it will be possible to propose adjustments to the methods, process and tools that can help build and manage better the process of developing games with agile methodologies and geographically distributed teams.

This work aims to understand and propose adjustments to this process by reporting a game construction case. For this, it is divided into the following three sections: the second one presents a bibliographic survey in order to better understand the concepts covered in the work; the third section describes the experience report: the game, the structure of the software factory and presents the process and tools that were used for development; finally, section four presents the reflections on the necessary adaptations for this context and final considerations.

2. A Review of Game Development Process by Agile Methods

Coram and Bonher [2005] say that Agile Methods offer a reasonable approach for the high degree of change and uncertainty in today's software development. Additionally, they also say that when these are combined with other agile principles, there can be a synergy that provides even more traction on the project goal.

The Manifesto for Agile Software Development prioritizes individuals and interactions, collaboration and responding to change, attending the class of software's project where the communication and monitoring change requirements are essential [Beck et al. 2001; Highsmith 2002].

On the other hand, develop game is a process of intense creation occurring in parallel to the project and has to be managed carefully. The creation is valuable but it is also significantly costly when not well managed [Keith 2010].

The Scrum was created in the last decade and stands out from other agile methods for greater emphasis on project management [Coram and Bonher 2005]. Gathers feedback and monitoring activities, in general, quick and daily meetings with the entire team, aimed at identifying and correcting any deficiencies and/or hindrance in the development process [Schwaber 2004]. Short development cycles, change management, collaboration and communication are essential.

Highsmith [2002] enforces that the agile movement covers a broader set of issues than the word methodology connotes: a chaordic perspective, collaborative values and principles, and barely sufficient methodology, reinforcing the importance of management the uncertainty of product requirements. In their study, Petrillo and Pimenta [2010] conclude that the adoption of agile practices in development of games can achieve promising results.

A previous adaption of Scrum that is based on the game development life-cycle is presented by Schild et al. [2010]. According to them, Scrum has strongly contributed to their project outcome. Based on these previous studies and considering the characteristics of our game project – small and distributed team's organization, a short development period of time (about 4 months) and the availability of customer to participate in the project meetings – Scrum was chosen as software development methodology. Others studies that applied Scrum in distributed teams [Paasivaara et al. 2008; Sutherland et al. 2009] also encouraged the use of Scrum.

3 Experience Report: The process of a social network game development

The project discussed in this paper was performed in a Software Engineering postgraduate discipline of Federal University of Pernambuco. Discipline goals were to instantiate a software factory, setting methods, processes and tools for development of a project. And the project chosen was the development of a social network mobile game, i.e., a game which runs in social networks (such as Facebook and Twitter) and in smartphones and tablets (with Android operating system). To setup and performs the project, the class had four months, from March to July/2011. Next we describe the game, named Catch the Pigeon.

3.1 The project

After several discussions, Catch the Pigeon game was specified and game design was build. The game is based on a carrier pigeon, which intend to deliver a message for the player by social networks. There are three kinds of carrier pigeons, one male, one fat and one female, each one with special characteristic, such as: more velocity, more lives and a merge of both. While the carrier flies with the message, several bad pigeons try to intercept him, colliding with him. Thus, the player goal is to protect the carrier pigeon by controlling him and by hitting the bad pigeons, in order to avoid collisions and preserves the carrier alive. Figure 1 shows a screen game play.

In addition to this general game play, the game has other main requirements: (i) the game should be playable as a Facebook game; (ii) the game should be playable as an Android application; (iii) the game should be playable with or with not internet connection, specially the Android version; (iv) when the user reaches a new level, the game must be able to post the message carried by the pigeon (written previously by the player), in the player Facebook and/or Twitter user account; (v) when the user reaches a new level, the game must save the player score; (vi) the game must build a high score board, and shows the board in the beginning of the game and in level transition screens.

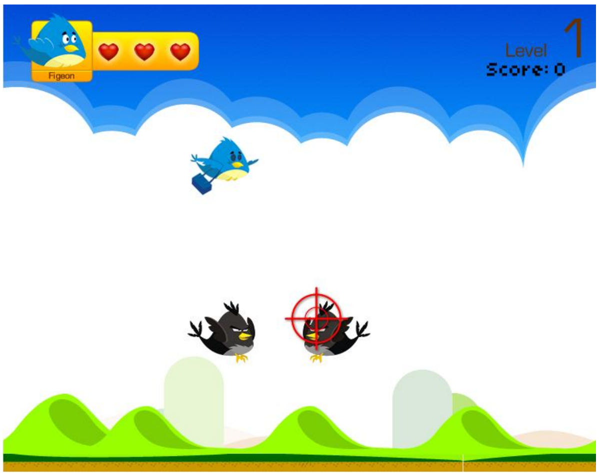

Figure 1: Catch the Pigeon game play

In addition to these requirements, other non function requirements were defined. There are: (i) scalability, in order to support possible increasing of players; (ii) security, in order to preserve social networks user's data; (iii) availability, which the goal is almost 100% of game player availability; (iv) performance, to permit game distribution be easily downloadable and game response times be adequate; and (v) maintainability, in order to increase productivity in development tasks. In addition, the maintainability is important to allow the extension of game to other possible social networks, such as Orkut or Google+, and other mobile platforms, such as iOS or Windows Phone.

3.2 The Team

The team was composed initially by 27 master's and doctoral candidates and, at the conclusion of the project, four months later, it had 23 members, supervised by 3 teachers and supported by 3 main external collaborators. As suggested by the agile Scrum methodology [Schwaber 2004], the team was divided into small teams. Each team had well defined roles but also complementary tasks as illustrated by Figure 2 that presents the factory's organization. The experimental factory was named MOSAIC (acronym of MObile and Social Applications in Cloud).

Stakeholders, Scrum Master (SM) and Product Owner (PO) are represented by professors and collaborators. The project management (PM) is executed by a post graduation student. The development support was divided in teams responsible for factory’s site, quality and process definition (PT – Process Team) and configuration management (CMT – Configuration Management Team). The Engineering teams were divided into game design (GDT), mobile development – that implement features of mobile game and Android tablet – and web development teams.

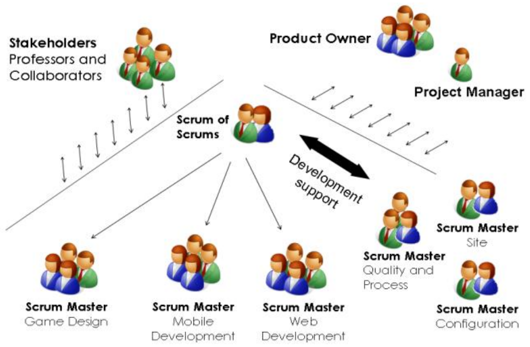

Figure 2: Factory organization.

3.3 The Process

The development process set for MOSAIC Factory is adapted from the concepts presented in “*Agile Game Development with Scrum”* [Keith 2010]. According to Keith [2010], each sprint must produce a playable version that should be refined until the final release to be launched on the market. Different from waterfall model, the activities detailed on Figure 3 are repeated in each sprint, producing a deliverable – a release of game. This proposal defines the following activities: concept, design, coding, asset creation, debugging, optimization, tuning and polishing. These activities in each sprint were also organized in order to enable parallel development.

The MOSAIC process also prioritizes agile methodology principles oriented to collaboration, iteration, change management and incremented development [Coram and Bonher 2005].

The MOSAIC process prescribes the following practices:

- First of all, an initial meeting established the set of functional and non functional items which represents the vision of the product to be delivered at the end of the project. This list is presented and negotiated with stakeholders, product owner and project manager to provide a common sense of what will be produced. The deliverable of this meeting is the product backlog. The product backlog will be divided in short periods of time – sprints – to improve the management of tasks completion and change requirements.
- Sprint: short period of time for completion of the selected items. In this experience, each sprint was held over two weeks (15 days).
- Meeting planning and items prioritization: meeting organized to define and identify which requirements will be developed at each sprint. The selection was made by product owner (based on his needs and priorities) and by the team (based on their availability, skills, and task force capacity). One of main deliverable is sprint backlog items. In this set was also considered the risk of each item. Some technological issues such as study of Facebook’ and Twitter’ API, modes of integration of different platform was selected on first sprints to reduce the risks of the project.
- Sprint planning meeting: detail and distribution of all tasks and activities of a sprint for each team, made on Tuesdays.
- Daily Scrum meetings: meeting involving the members of each team. Each team established the agenda of its meetings. This provides the discussion of impediments, the progress of tasks, and general discussions about technological or other issues between the participants of a specific team.
- All Thursdays nights, a scrum of scrum was conducted to promote exchange information between teams.
- Sprint review meeting: to analyze the results obtained during the time of Sprint, including all team members, held on Sunday and some sprints on Monday. When new activities and requirements were raised, these were added in the Product Backlog to be developed in the next sprints.
- Retrospective: the goal of this meeting is to reflect on the events and the effectiveness of the team that worked together during last sprint. It took place after the sprint review meetings.
- Improvements: every improvement identified during the course of the project was included in project backlog for later sprints.
- Launch: delivery of the version of the game with the improvements produced each sprint.

3.4 Activities performed in each sprint

According to the specific features of the project, game on mobile devices, social networks, geographically distributed team, the following activities were planned (See Figure 3):

- Concept: where ideas are generated and tested, using prototypes for evaluation;
- Design: Design of ideas, the feasibility study defining tasks and design features that will be implemented and also the tests were defined to evaluation of requirements;
- Asset Creation: detailed specification of the game, art-related components (figures of the characters, scenarios and other), sounds of the game (when a character are achieved or die) and animations (explosions and others);

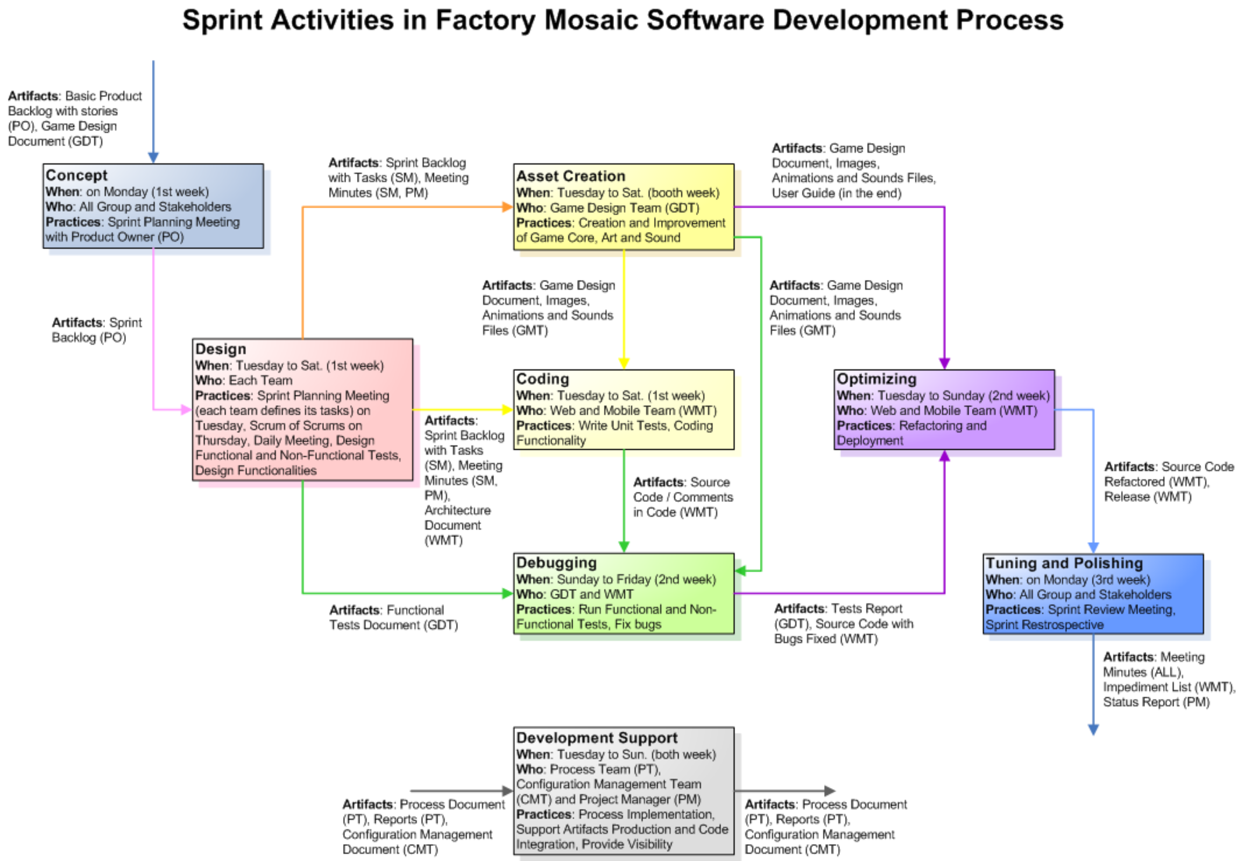
Sprint Activities in Factory Mosaic Software Development Process
Artifacts: Basic Product Backlog with stories (PO), Game Design Document (GDT)
Concept
When: on Monday (1st week)
Who: All Group and Stakeholders
Practices: Sprint Planning Meeting with Product Owner (PO)
Artifacts: Sprint Backlog (PO)
Artifacts: Sprint Backlog with Tasks (SM), Meeting Minutes (SM, PM)
Asset Creation
When: Tuesday to Sat. (booth week)
Who: Game Design Team (GDT)
Practices: Creation and Improvement of Game Core, Art and Sound
Artifacts: Game Design Document, Images, Animations and Sounds Files, User Guide (in the end)
Artifacts: Game Design Document, Images, Animations and Sounds Files (GMT)
Artifacts: Game Design Document, Images, Animations and Sounds Files (GMT)
Design
When: Tuesday to Sat. (1st week)
Who: Each Team
Practices: Sprint Planning Meeting (each team defines its tasks) on Tuesday, Scrum of Scrums on Thursday, Daily Meeting, Design Functional and Non-Functional Tests, Design Functionalities
Artifacts: Sprint Backlog with Tasks (SM), Meeting Minutes (SM, PM), Architecture Document (WMT)
Coding
When: Tuesday to Sat. (1st week)
Who: Web and Mobile Team (WMT)
Practices: Write Unit Tests, Coding Functionality
Optimizing
When: Tuesday to Sunday (2nd week)
Who: Web and Mobile Team (WMT)
Practices: Refactoring and Deployment
Artifacts: Source Code Refactored (WMT), Release (WMT)
Artifacts: Source Code / Comments in Code (WMT)
Debugging
When: Sunday to Friday (2nd week)
Who: GDT and WMT
Practices: Run Functional and Non-Functional Tests, Fix bugs
Artifacts: Functional Tests Document (GDT)
Artifacts: Tests Report (GDT), Source Code with Bugs Fixed (WMT)
Tuning and Polishing
When: on Monday (3rd week)
Who: All Group and Stakeholders
Practices: Sprint Review Meeting, Sprint Restrospective
Artifacts: Meeting Minutes (ALL), Impediment List (WMT), Status Report (PM)
Development Support
When: Tuesday to Sun. (both week)
Who: Process Team (PT), Configuration Management Team (CMT) and Project Manager (PM)
Practices: Process Implementation, Support Artifacts Production and Code Integration, Provide Visibility
Artifacts: Process Document (PT), Reports (PT), Configuration Management Document (CMT)
Artifacts: Process Document (PT), Reports (PT), Configuration Management Document (CMT)

Figure 3: Details of process activities performed in each sprint.

- Encoding: encoding according to the design established in previous activity. Example: codification of actions related to the flight of the characters in each of the platforms (mobile and web). The TDD (Test Driven Development) was applied;
- Debugging: to minimize the occurrence of errors in features implemented. Each sprint has an activity of application of tests, identifying and fixing bugs. Correcting errors is part of the work that needs to happen before the resource is delivered in each sprint;
- Optimizing: this phase aims to improve the application incrementally, refactoring and fixing any problems related to non-functional requirements, revise requirements for graphics, artificial intelligence and others;
- Tuning and polishing: at this stage, the design: features of the game, for example, can be refined to improve the interaction in the game.

3.5 Artifacts and Project Management Tools

In addition to regular Scrum artifacts, several specific artifacts, related to the nature of the application that was developed, were included. A web site supports the publication of project artifacts. The following artifacts were produced:

- Process' document: that details the MOSAIC Factory development process addressed to developers, collaborators, external community;
- Configuration management' document: that details the process of tracking versions, supporting and auditing the change settings;
- Game design's document: all the details of game design and the incremental improvements that has been made;
- Architecture's document: overview of architecture – its components, interfaces;
- Test cases' document: usability, playability, functional and non-functional test cases;
- Meeting minutes: allows the member of each team and also the stakeholders access to meeting minutes;
- Status reports: produced each sprint to provide details of the development progress;
- Product backlog: functional requirements and non functional requirements defined by user stories as recommended in literature;
- Sprints backlog items: each selected backlog item with its prioritization;
- General guidelines;
- Tasks: all the tasks, its assigned member, prioritization, duration;
- Lists of impediments;
- Source code and comments;
- Bug reports;
- Images, animations, sounds and releases of the game.

To support internationalization, all activities have been documented in English.

The tools were selected based on distributed teams adopted practices and in their capability to contribute with management, development, tests and communication in the project. The selected tools were:

- FireScrum [2011]: used initially as a project management tool, it implements the main concepts of Scrum;
- Mantis [2011]: used as a bug tracker's tool. It was adopted by the project management since it provides more detailed control of each requirement and associated task. A document that contains the main guidelines for use of Mantis was prepared and made available in the document area of the MOSAIC factory's site;
- TestLink [2011]: used as test management tool, due to convenience in defining the test cases;
- GitHub [2011]: the repository from GitHub made possible the collaboration in developing the application code;
- Wordpress [2011]: an easy to use Content Management System (CMS) to the site of the factory.

During development phase, features such as task board, posts on the walls, and whiteboard were also used for task management.

3.6 Steps and details of development's issues

An initial sprint (called sprint 0) was held aiming to organize the factory structure, draw up a preliminary document of game design, carry out studies related to technologies to be used, and choose support tools and the preliminary definition of the development process.

The development phase effective started on April. It was planned to run in five sprints of two weeks duration (each one). In the end, a first release version of the game was provided. The development sprints were conducted in accordance with the schedule below (Table 1):

Table 1: Sprints' details

Sprint	Date	Goal
1	April,25 to May,08	A game with a character and three enemies. The game begins and if the character reaches the end of the screen, the message is sent to Facebook.
2	May,09 to May,22	The game with two stages. Attack of the enemies. Post a message on Facebook.
3	May,23 to June,05	Integration between the web application, mobile applications and server (interface, authentication, authorization, post, get rated and top five).
4	June,06 to June,19	Execution of tasks of fixes in application (web, mobile and server) to review and refine the previous sprints items pending.
5	June,20 to July,03	To-do Implement previous sprint and provide a version with four phases and more a character (leader of the Evil Pigeons clan).

During the Sprint 3 a Game Jam was developed. The event was attended by 22 members over the weekend, working about 36 hours. During this period, 77 tasks were resolved. A new Game Jam was held on June. In this event, tasks related to the final game's release to be launched were completed

Game Design Team began their activities earlier to provide a graphical preview of the game for all participants. Before the sprint 1, it was developed a first draft of the game design document, which contains the description of the core of the game and the game's script. In parallel to this task, the team also implemented the first images of the game, such as scenarios, characters and others. This design has been continuously improved in each sprint, adding sounds and animations as explosions and others. In these tasks, the team had the collaboration of an external designer. After the first features have been implemented, the team organized a sub-team to draft and implement the functionality, usability and exploratory testing scripts.

Mobile development team worked to implement features for mobile game and Android tablets. During the sprints, this team faced more challenges because some members did not have coding experience to Android. In this context, it was necessary to take time to study the technology and then implement the code. Pair programming technique was also applied, joining an experienced programmer with a beginner programmer, to overcome this obstacle.

The Web development team was responsible for implementing features related to the web application. The web application consists of a game that is accessed by Facebook. The team started developing the gaming functionality, such as opening screen, character selection screen, the screen transition level. Flash Builder was used to achieve so. The game engine was implemented with the algorithm of persecution and features such as collisions that implement the death of a bad pigeon, when it coincides with a user click and the death of Messenger pigeon, when it was hit by the bad pigeons a certain number of times. Pair Programming was also applied.

Initially this application was implemented separately from the mobile application that uses Java in its development. Some features could be refactored to correspond with the features implemented in mobile version. Likewise, features implemented on the mobile version have been refactored according to web implementations.

The urge to create a specific group to work in the server integration maintenance was identified during the development of the features. Then the web engineering team has created a sub-team, with three developers to implement the server-related features. In the final sprint, this team received contributions from developers of Mobile engineering team to ensure that integration with mobile applications has been well implemented.

These features have proven to be more complex. In addition to implementing the integration between web and mobile applications, the server needed to implement security solutions, while preserving the minimum information of the user and scalability, to ensure the operation even with a large number of user requests.

Tests were conducted at each sprint, including usability, exploratory, functional and non-functional tests. TDD tests were applied in mobile, web and server implementations (infrastructure services, persistence and communication). Testing activities were designed and conducted by a sub-team within the game design team, with the participation of the process team. In each sprint, the functionality implemented was tested and also regression tests were conducted. The tests were fundamental to the improvement of applications, since they pointed out the quality issues that need to be resolved. Tests have been reported in TestLink and bugs were reported in Mantis.

3.7 Managing projects

The project management activities in MOSAIC Factory had the focus on the following items:

a. staff: this activity included all members regularly enrolled in the of Software engineering class. All students had their profile parsed through questionnaires. In addition, all members had the opportunity to indicate the area they had more affinity. The test of Myers-Briggs Type Indicator (also known as test MBTI) [Inspiira 2011] was simplified and applied in order to identify how the members perceived the world, and how they would make decisions. The main development areas were identified and, as presented before, four different teams were created to allocate all members: Game Design, Engineering (Mobile and Web), Process and Configuration management. Each team had their own team leader that was chosen on the basis of tests and profiles.
b. planning project: all the planning of the project was conducted with all team leaders and the owner of the product, based on agile methodologies. There was a weekly meeting scheduled to track down all activities performed by each team.
c. feasibility of infrastructure: the factory site was hosted in a Private Virtual Machine (VPM) to make available all the documentation to team and external collaborators. Furthermore, all development and support tools were hosted in VPM which included the change request tool (Mantis), the testing framework (TestLink), Java container (Tomcat) and others.
d. impediments solutions: a project management activity was related to the solution of obstacles. The items necessary for the progress of other dependent task should be requested for a member of a team, team leaders and/or to the project manager.

4 Lessons Learned and Final Considerations

This paper has shared the experience in development of a mobile and social game named Catch the Pigeon by MOSAIC Factory, an experimental software factory developed within the discipline of Software Engineering from post graduate course of Informatics Center in Federal University of Pernambuco. This experience was fundamental to learn in practice various concepts of Software Engineering that can be used to develop mobile and social applications. This knowledge will also contribute to solving other problems in Software Engineering area.

Here are presented lessons learned in context of MOSAIC Factory experience:

- Factory organization in teams contributed to the project approach, but it was also important, in some stages, to move developers from one to another team to help in activities with more complexity. This demanded a multidisciplinary team, prepared to solve different challenges with different technologies;
- Planning poker process was used to estimate the effort. Since, there was no record data about the productivity of team yet, the estimates were being more precisely only after the third sprint. Considering that the members had other activities in parallel, in some weeks the dedication to the project was lower than in others, which made it difficult to keep track of estimate delivery date of the requirements based on effort on the task and deadlines;
- At the beginning of the factory team members focused more on the product to be developed. The activities to support development were implemented in parallel to development, which somehow ended up causing errors, conflicts and reducing the productivity, especially in the initials sprints. Support activities like communication tools, repository and others had to be implemented first;
- Process was initially set at a high level view. Then, during the development, it was detailed with contributions from developers. This dynamic has resulted in a process that was adjusted to the real demand of the teams;
- Considering Distributed Software Development (DSD) remote communications tools (like Skype, Gtalk and mail-list), collaborative productivity tools (like GoogleDocs and Dropbox), and the frequent update of documentation and artifacts at web site, the factory team communications needs overcame geographic distribution;
- Negotiation was crucial during the development of the project to remove impediments and manage tensions that arose within the groups. It is also important to decide the requirements that made

possible to enrich the game, methods and tools to be used.

● Scrum is well suited to game development since it support the change management, iteration, collaboration necessary in a process of intense creation.

● Conventional process must be adapted to incorporate the activities of design, asset creation, coding, tests and improvements in each sprint. A game project differs from others applications due to more emphasis in these activities related to creation of scenarios, characters, effects and also metrics, tools and processes to usability and playability management.

Although the results obtained in this experience are likely to be applied in other contexts, it is necessary to consider that the generalization of these results represents a limitation of this work, mainly because it was conducted in an academic environment, which differs from industrial environment, even having sought to approach the most of industry features.

As future works, we suggest a replication of this experience in the context of another team in order to observe if the solutions pointed here are also applicable, to establish a more formal set of practices to develop games, with social and mobile characteristics.

Acknowledgements

We would like to thank to Informatics Center (CIn) of Federal University of Pernambuco (UFPE) and to Recife Center for Advanced Studies and Systems (CESAR) which provided the infrastructure to development and the devices for tests, mainly during the Game Jams.

Thanks to Felipe Furtado, Luiz Borba, Alvaro Cavalcanti, Tairone Cesar, Nelson Glauber, Rodrigo Assad and Elias Queiroga for help us with technical and conceptual questions. Janiel Almeida who acted as external collaborator with game design. Professors Silvio Meira, Vinicius Garcia and Jones Albuquerque who stimulated, accompanied, required and helped us several times.

Finally, we thanks each member of the Factory Mosaic's team: Aline Ayres, Angela Peres, Dhiego Abrantes, Fernanda Martins, Fernando Selleri, Georgia Silva, Ivan Samuel, Jamilson Antunes, João Emanoel, Kellyton Brito, Lenin Otero, Leonardo Araújo, Lubnnia Morais, Marco Andre, Marcone Ferreira, Paulo Fernando, Rafael Roballo, Rafael Rocha, Raoni Kulesza, Rodolfo Santos, Tâmara Duarte, Rachel Pernambuco and Thiago Jamir.

References

BECK, et al., 2001. Manifesto for Agile Software Development [online]. Available from: http://www.agilemanifesto.org/ [Accessed 28 July 2011].

CORAM, M. AND BOHNER, S., 2005. The Impact of Agile Methods on Software Project Management. *In: Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 4-7 April 2005 Greenbelt*. Los Alamitos: IEEE, 363-372.

FIRESCRUM, 2011. *FireScrum: The Open Source Scrum Tool* [online] INES. Available from: www.firescrum.org [Accessed 10 September 2011].

GITHUB, 2011. *GitHub: Secure source code hosting and collaborative development* [online]. Available from: github.com [Accessed 10 September 2011].

HIGHSMITH, J., 2002. Agile project management – creating innovative products. Boston: Addison-Wesley.

INSPIIRA, 2011. *Free Personality Test* [online]. Available from: www.inspiira.org [Accessed 11 April 2011].

KEITH, C., 2010. Agile Game Development with Scrum. Boston: Addison-Wesley.

MANTIS, 2011. *Mantis Bug Tracker* [online]. Available from: www.mantisbt.org [Accessed 10 September 2011].

MARTIN, K. AND HOFFMAN, B., 2007. An Open Source Approach to Developing Software in a Small Organization. *IEEE Software*, 24(1), 46–53.

PAASIVAARA, M., DURASIEWICZ, S. AND LASSENIUS, C., 2008. Distributed Agile Development: Using Scrum in a Large Project. *In: Proceedings of the 2008 IEEE International Conference on Global Software Engineering (ICGSE '08), August 17-20, 2008*. Washington: IEEE Computer Society, 87-95.

PETRILLO, F. AND PIMENTA, M., 2010. Is Agility out there? Agile Practices in Game Development. *In: Proceedings of the 28th ACM International Conference on Design of Communication (SIGDOC '10), September 26-29, 2010, São Carlos/SP*. New York: ACM Press, 9-15.

SCHWABER, K., 2004. Agile Project Management with Scrum. Redmond: Microsoft Press.

SCHILD, J., WALTER, R. AND MASUCH, M., 2010. ABC-Sprints: adapting Scrum to academic game development courses. *In: Proceedings of the Fifth International Conference on the Foundations of Digital Games (FDG '10), June 19-21, 2010, Monterey*. New York: ACM Press, 187-194.

SUTHERLAND, J., SCHOONHEIM, G., KUMAR, N., PANDEY, V. AND VISHAL, S., 2009. Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India. *In: Proceedings of the 2009 Agile Conference (AGILE '09), August 24-28, Chicago*. Washington: IEEE Computer Society, 277-282.

TESTLINK, 2011. *TestLink: Test Management Tool* [online]. Available from: www.teamst.org [Accessed 10 September 2011].

WORDPRESS, 2011. *WordPress: Blog Tool and Publishing Platform* [online]. Available from: wordpress.org [Accessed 10 September 2011].