Process Control in the Development of Game-Based Learning Environments

Antti Kirjavainen
University of Jyväskylä
P.O.Box 35
40014 University of Jyväskylä
Finland
antti.kirjavainen@iki.fi

ABSTRACT

This paper examines the development process of digital game-based learning environments (GBLE), focusing on the process control. The aim is to explore the alternatives to process control in GBLE development and provide guidelines for planning and carrying out process control in GBLE projects. This study is a part of the research project Human-Centered Design of Game-Based Learning Environments. The overall aim of the project is to construct a multidisciplinary and user-driven process for the development of digital GBLEs. The study was conducted according to the principles of development research and action research. The action research cycles consisted of four game development projects. The results of the study indicate that GBLE development has many uncertainties that advocate the use of empirical process control. The study also provides guidelines of using empirical process control in concert with human-centered design, game design and learning-goal oriented game concept creation in GBLE development.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*Software process models*

General Terms

Design, Management

Keywords

Game-Based Learning Environments, serious games, game development, development process, process control

1. INTRODUCTION

In order to develop high quality game-based learning environments (GBLE), the development process of those environments needs to be studied and a process model and guidelines for quality development must be constructed. This study focuses on the process control.

Project management has three main responsibilities: making a plan to deliver the desired product, monitoring the work and reacting to emerging problems ([6], [12]). These three responsibilities are combined in process control ([7], [11]).

In earlier research ([2], [5]) processes of GBLE development were elicited, modeled and guidelines for quality development process of GBLEs were drawn. Otherwise there are not very much literature on how to ensure quality in GBLE development by means of process control, although there are some studies that suggest guiding the design process with feedback on testing user interfaces, either by subject matter experts [13] or by end-users [1]. Therefore in this study the literary basis for the research is derived from software engineering.

The two main approaches of process control in software engineering discipline are defined process control and empirical process control. Defined process control is presented in the traditional software project management literature [11]. Empirical process control is implemented in the more iterative software process models such as the spiral model and the Scrum process model [10]. These two process control models are discussed in chapter 3.

2. METHOD

This study is part of the research project Human-Centered Design of Game-Based Learning Environments. The overall aim of the project is to construct a multidisciplinary and user-driven process for the development of digital GBLE. The study was conducted according to the principles of development research [15] and action research. The action research cycles consisted of four game development projects.

In these projects varied sets of methodologies were applied, including different approaches in process control. The action research approach was based on the researcher being an active participant in the planning of the case projects and also a stakeholder, consultant or a member of the development team in each of the projects. Other members of the research group also participated in a similar way. This viewpoint to the research was contrasted by studying each project as a case study, using the project documentation and interview data to build an alternative view of the project.

The research questions are:

1. What kind of process control model is most advantageous and feasible to the development of GBLE (both

Ace 2009, Oct 29-Oct 31, 2009, Athens, Greece

in general and especially in those projects concentrating on game design and human-centered design) and

2. what kind of adaptation of that process control model would lead to a high-quality development process in this domain.

This study contrasts the results of other studies in the field with the results acquired in the development projects and draws conclusions on the impacts of different kinds of process control models on the development projects. The second part of the study investigates how the selected process control model could be adapted to the development of game-based learning environments.

2.1 Data collection and analysis

The data was gathered through observation, interviews, and project documentation. The case studies were observed during the diverse development phases. Observations were recorded in research diaries. The participants were interviewed both individually and in groups after the development projects. The individual interviews consisted of open themed questions. In the group interview sessions historical mapping technique was utilized to construct a shared map of the project situation, events and issues during the project and changes that occurred in the project.

The project documentation consisted of project plans and reports (e.g. phase plans and reports, weekly project diaries). The project documentation was examined for statements about process planning, project activities, work products, events during the project as well as monitoring and project management actions that were carried out. The plans and design documents that the projects produced were examined for indications of impact of the process on the projects' outcome.

The project review reports to which several stakeholders of the project contributed were examined for statements on the teams' process the impact of it in the project. The development teams of the projects were also interviewed about these matters.

3. PROCESS CONTROL IN SOFTWARE ENGINEERING

In software development the process control is achieved through project management, through activities such as project planning, scheduling, project monitoring, measuring, controlling, inspecting and reviewing ([12], [14]).

As mentioned in the introduction, there are two main approaches of process control in software engineering discipline: defined process control and empirical process control. Defined process control is presented in the traditional software project management literature [11]. Empirical process control is implemented in the more iterative software process models such as the spiral model and the Scrum process model [10].

It can also be said that the distinction between defined and empirical process control is not necessarily a binary one but instead more of a continuum. In the other end of the continuum there is pure defined process control and on the other pure empirical process control. Sequential software process models rank at the defined end of the continuum and iterative process models are placed further towards the empirical end the shorter their feedback cycle is.

In this study these two approaches to process control are evaluated for GBLE development. The following chapters discuss each approach in light of software engineering and, where applicable, game development literature.

3.1 Defined Process Control

Defined process control means that every task has been defined as a process and the inter-relation of these processes has been defined in detail [11]. The control comes from predictability, meaning that with the same inputs the defined processes should produce the same results in different implementations. The manager can plan the whole project before it starts and knows what resources it takes to complete [12]. The advantage of this is the predictability: the decision of undertaking a development project can be made with the help of estimations of resources needed and the benefits that the end product provides. The drawback of this is that if the processes are not explicitly defined, the plans and estimations are not reliable [12].

The software project is first initiated and its scope is defined via requirements elicitation tasks. The initiation includes the feasibility analysis task in which the possibility of completing the project with available resources is estimated. Then the project itself is planned in the form of hierarchical decomposition of tasks, the associated deliverables of each task are specified and characterized in terms of quality and other attributes in line with stated requirements, and detailed effort, schedule, and cost estimation is undertaken [14].

In the ongoing software project change is managed with the following processes: risk management, quality management and plan management. In risk management risks threatening the success of the software project are identified, estimated and plans for avoiding and minimizing them are made. In quality management, the quality is defined by terms of attributes of the project and its end products. The desired quality attributes of the end products are specified in requirements. The procedures for on-going quality assurance processes are defined in quality management.

Plan management involves directing, monitoring, reviewing, reporting and revising adherence to the project plans. Plan management is primarily responsible for responding to changes in the project environment. The project process is also monitored throughout the project. Adherence to plans, outputs and completion conditions for each task are analyzed. Deliverables are evaluated for their adherence to required characteristics. Resource use is also monitored. When the monitoring indicates that adherence to the plan is at risk, process control is responsible for making changes in the project plan. [14]

As the end product quality evaluations are tied to completion of the said products the process monitoring and control cycle is related to the completion of the end products. In a linear process model this can mean that failure to fulfill the desired quality attributes of the end product is only found out after the development phase. This and the lack of instruments to control major change in the midst of the project means that defined process control model does not yield good results in projects with a lot of uncertainty and change.

3.2 Empirical Process Control

Empirical process control models have been developed for processes that are highly unpredictable. They provide control through frequent inspection and adaptation for processes that are imperfectly defined. Empirical process control models have been adapted to the software engineering discipline from the engineering process discipline. [11] Many agile software development methodologies such as OpenUP, Crystal and extreme Programming use an empirical process control model.

In this study Scrum methodology is examined as an example of empirical process control. Scrum is a software process model that focuses on project management ([8], [9]). Scrum has been noted to be effective in software development projects were there was much uncertainty and change expected [10]. The reason that Scrum was chosen is its focus on the process control and the fact that Scrum was the methodology that made empirical process control methodology widely used in software engineering.

One of the principal tenets in Scrum is that everyone on the team is involved in the process [4]. A scrum team is typically made up of between five and nine people [11].

The product owner is the project's key stakeholder and represents users, customers and others in the process. The product owner is the person who is responsible for adding items to the product backlog and re-prioritizing the items in the backlog. The product backlog is a prioritized features list containing every desired feature or change to the product. All additions and changes to the backlog must go through the product owner. This helps the scrum team deal with changes in the project. [11]

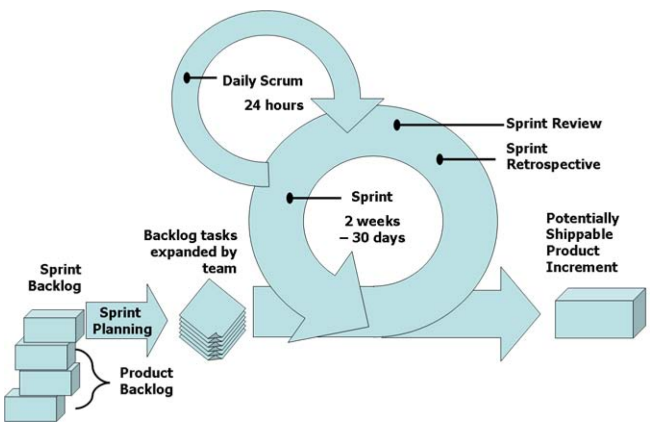

Figure 1: Diagram of the Scrum process.

The Scrum Master is responsible for making sure the team is as productive as possible. The Scrum Master does this by helping the team use the Scrum process, by removing impediments to progress and by protecting the team from outside pressure during Sprints [11].

Scrum breaks down production into short work cycles called Sprints [11] (Fig. 1). Every Sprint produces a visible, usable, deliverable product [10]. Before every Sprint the development team sets its own goal for the Sprint and commits to completing it [11]. A sprint is timeboxed development, meaning that the end date for a sprint does not change. The team can reduce delivered functionality to achieve the sprint goal in schedule. [10]

At the start of each sprint, a sprint planning meeting is held during which the product owner prioritizes the product backlog, and the scrum team sets their Sprint goal, including selecting the work they can complete during the coming sprint. That work is then moved from the product backlog to the sprint backlog, which is the list of tasks needed to complete the product backlog items the team has committed to complete in the sprint. [11]

Each day during the sprint, a brief meeting called the daily scrum is conducted [10]. All team members are required to attend the daily scrum. In daily scrum every team member presents work done after last meeting, what she will do before next meeting and possible obstacles that hinder her work. [11]

At the end of each sprint, the team demonstrates the completed functionality at a sprint review meeting, during which, the team shows what they accomplished during the sprint. [11]

The main advantages of Scrum are 1) the increased capability to deal with change and 2) the shortened feedback cycles in development. The Scrum process combines flexibility and freedom to pragmatic assessment of real work done as well as the process itself [4].

Stakeholders can redirect project's goals between Sprints. Because the Sprints are short work cycles, this rarely results in a lot of wasted works. [4]

Iterative development is stressed in Scrum. The focus on producing shippable versions of the product means that the project's stakeholders can base their assessments and decisions on concrete products, not just documents, specifications, plans or estimations. [4]

With Daily Scrum meetings the progress and possible deviations from normal progress become visible to every team member including the project manager (scrum master) [10]. In game projects Scrum has the benefit of enabling game designers see complete pieces of finished functionality in regular intervals [4].

The potential drawback of empirical process control is that there are no projections at the start of the project on how much resources, including time, are needed to complete a finished solution for the problem in question with the desired quality attributes. This drawback is lessened by the facts that as the project goes on this estimation is made possible by constant monitoring of the project's progress and that if the project contains much uncertainty and is subject to change this kind of estimation could not be done accurately with any other methods.

4. RESULTS

The results of this study are presented as follows: first the case projects are described along with their characteristics related to process control and experiences noted related to the chosen approach in process control in each of the projects. After that the key activities in GBLE development, identified by previous study, are discussed in the light of the two process control models. Based on these results conclusions are drawn into the benefits and drawbacks of the two process control models compared in the study for GBLE development. Finally, guidelines are drawn for combining empirical process control, exemplified by the Scrum process, with the key activities of GBLE development.

4.1 Process Control in Case Projects

In this chapter the four case projects are described in terms of goals and overall process. Their process control

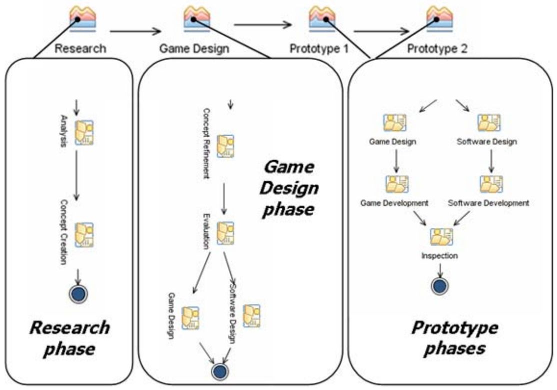

Figure 2: Diagram of the phases and activities in the Gameli project.

model is then discussed and the experiences of the chosen process control model and its impact on the project are described.

Gameli V.1

The aim of the Gameli V.1 project was to develop an engaging learning game on top of the GameWorld modeling and simulation software developed by Centre of Information Technology for Education of the University of Hong Kong.

The project used an iterative modification of the waterfall process model where the design, implementation and testing were done in two iterations (Fig. 2). The project also used game concept and design documents to structure the game design task. These documents were used as a basis for requirements specification which in itself consisted of a specification document with prioritized requirements.

The process control model in the Gameli project was defined process control. Each phase of the project had planned activities which were expected to have the assumed results and a set schedule. At the end of each phase the work of the project was inspected by the steering group for compliance to the plan. The plans for later phases were revised where applicable informed by the experience on the activities gained from project work but this was limited by the strict planning before the project and limited resources available in the middle of the project.

The project team members noted that many of the original estimations of the workloads of the planned activities were found as inaccurate: *"Most notable risk in this [game design] phase was the tight schedule. Calculated with planned resources the risk did occur."* [1]

In addition to that the early concept creation and subsequent HCD evaluation activities called for change in the project plans as well as requirements which had to be discarded as the project had not planned to deal with that much change and did not have the resources to implement the change demands when they were encountered. Small changes were made to the project plans where possible but the project reviews indicate that neither the project team members nor other stakeholders were satisfied with the project's ability to change its course in the midst of the project.

[1] From Gameli V.1 Game Design Phase Report, translated by author

This inability to deal with change could probably have been lessened by some degree with risk management but these experiences indicate that the inherent drawback of defined process control of not coping with uncertainty or change is also present in GBLE development projects.

Gameli V.2

The goal of Gameli V.2 project was to expand on the earlier Gameli V.1 prototype and to produce a simulation game environment/simulation game design environment that could be used in classroom in natural sciences education.

In this project the experience from earlier projects was taken into account when deciding the development process and the methods used. This project used the same kind of process model as Gameli V.1 project: an iterative modification of the waterfall process model where the design, implementation and testing were done in two iterations (Fig. 3). The focus however was more in the iterative and prototyping nature of the model: the requirements phase used rapid prototyping to learn about the product that was being developed and the first implementation iteration was light enough to allow for significant changes after testing. This was done also to allow for more significant role for users in the development process.

The second Gameli project also used the defined process control model. The process design was altered slightly to allow for more flexible adaptation to changing requirements and findings during the HCD activities by defining the goals for each phase in a more general manner. This allowed the team to expand on the goals after they had the results from HCD activities: *"The information gathered at the [HCD] teacher workshop vas extremely valuable in specifying the requirements."* [2]

This however lead to more uncertainty as the team was not sure if they had resources to complete the goals with the activities planned because the activities had been planned before the goals were specified in high fidelity. *"The change control was certainly an issue at that moment. There were a lot of possible routes to take and it was evident that everything could not be included."* [3]

This uncertainty did not manifest itself in actualized risks as the project group was able to prioritize the goals they set for each phase and complete the at least the goals that were crucial for the completion of the project but still the risk of not having enough resources to complete crucial goals was present.

The Social Responsibility Game

The goal of the Social Responsibility Game (SR Game) was to produce a game design for a game that would communicate the social responsibility aspects of a Agora Game Lab's partner organization.

In this project the focus was solely in game design. This allowed game concept generation and design methods to be taken into close inspection. The overall process model used was a sequential model, although this time consisting only of the concept generation (or requirements) and design stages

[2] An excerpt from the Analysis Phase Report, translated by author

[3] A comment from a developer in the post-mortem interview, transcribed and translated by the author.

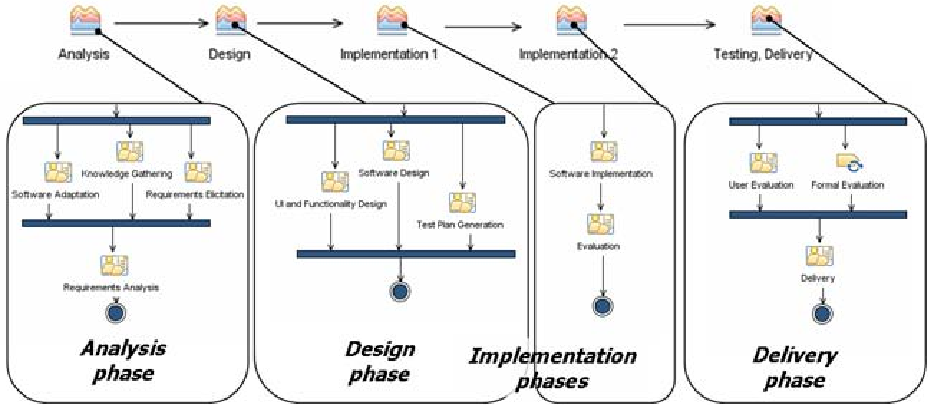

Figure 3: Diagram of the phases and activities in the Gameli V.2 project.

(Fig. 4). Game design documents were used to document the game concepts generated and the game design work done throughout the project. The methods used in game concept generation included brainstorming about the game's subject matter, harvesting ideas from existing games, using game design patterns as inspiration and creating competing game concepts. The game concepts were evaluated by testers from the target user group of the game as well as expert-evaluated. The game design was based on using the game design document but included also gameplay prototypes and expert evaluation.

The process control model of the project was a mix between defined and empirical process control. The project had a fixed set of phases with distinct responsibilities, but the activities were not planned in detail beforehand. Instead the project group used the phases as iterations to expand the game design work and explore the different possibilities and used the inspections between phases for feedback and advice on the planning and goal-setting of the next phase. In the end a game design document was produced and in fact the project had time to revise the created document for one increment before ending the project.

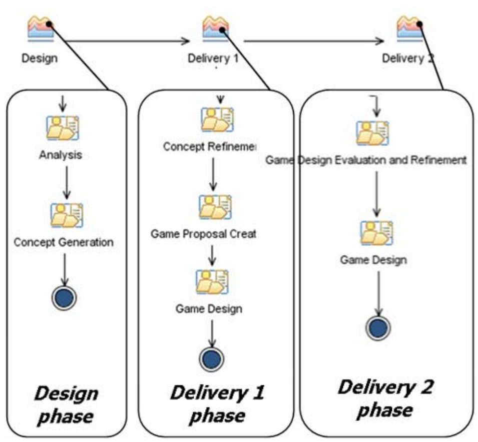

Figure 4: Diagram of the phases and activities in the Social Responsibility Game project.

The project reviews indicate that all stakeholders considered the project successful. The project team members commented that they could have validated the design decisions made in the process better if they had made and evaluated more gameplay prototypes during the project: *"The prototyping of the game should be started as early as possible."*[4] This argument was supported by some of the experts that reviewed the project.

The Peatland Adventure

The Peatland Adventure is a web-based adventure game designed to support in learning natural science related to the peatland nature. It is designed to be used both in the school context and as a stand-alone game outside school for various user groups. It is a part of the Virtuaalisuo (Virtual Peatland) virtual environment (http://www.virtuaalisuo.fi/). The focus in the design of the game was to make a motivating and interesting game that would provide the players with positive attitudes towards the peatland nature and environment in addition to providing the content-specific knowledge.

The development of this game started as a part of the development of the Virtuaalisuo virtual environment development. The original pre-production phase was very long as the whole environment was specified at the same time. After the first analysis phase the game project was separated as a project of its own. The latter part of the project used a different approach to the project process: It experimented on a more flexible team structure that started as a small two-people concept design team and expanded when the project proceeded to further phases. The project benefited from a lengthy concept generation phase that preceded the project. The concept generation phase included brainstorming possible game concepts to develop and testing and re-developing them with user participants. The development process had features from both sequential and cyclical development model (Fig. 5). One thing that made the project unique was the game genre: being an adventure game, the basic game mechanics did not involve a lot of design work. Instead, the level and game story design took most of the game design time. This allowed the team to adopt a kind of cyclical process where the game was developed one level at a time after the overall user interface and game look had been established.

[4]Comment from an interview with one developer, translated by author.

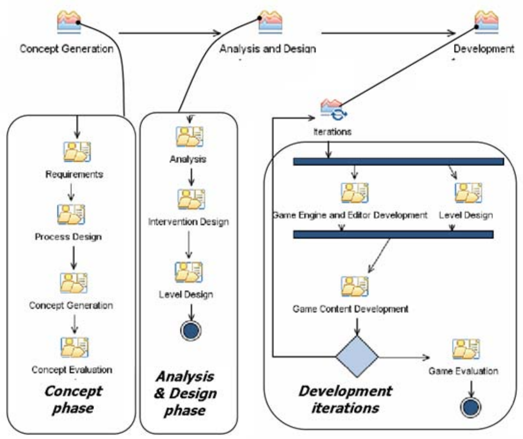

Figure 5: Diagram of the phases and activities in the Peatland Adventure project.

The Peatland Adventure project used different manifestations of the empirical process control model. The first, used in the original requirements phase, was a model driven by the HCD process. The idea was to iteratively elicit requirements for the game and create game concepts that were assessed by the HCD design collaborators. This phase was terminated when the concepts created in the final iteration were assessed as not fulfilling the requirements gathered.

The second part of the project started with a game concept and learning goal definition on top of which the project was founded. The project group worked autonomously to refine the game concept and design to confirm to the requirements set in the first part of the project. The project work was iterative and empirical and the project group was in control of deciding what to work on in each iteration. This approach lead to the experience that development iterations were also a learning experience: *"The first level design and learning goal and content design iteration served as a learning experience for us designers."* [5]

The process control was close to Scrum but did not adhere to the specific activities of the Scrum project model. Also in contrast with other projects in this study, HCD methods were not used during the development work of this project.

The self-directed and iterative manner of working was noted as effective and rewarding by the project team members. The team members also commented that they were learning very efficiently about different aspects GBLE development while working together as a tightly collaborating and self-directed project team. In the project report other stakeholders also commented on the high quality of the finished game and the ability of the project team to produce the game in a relatively short time.

4.2 Examination of Key Activities

In earlier research ([2], [5]) processes for GBLE development were constructed. This part of the result description of the study discusses the key activities in the GBLE development process and the impact of those to the process control of the project. The three key activity groups identified in the previous research [2] discussed here are 1) GBLE concept generation combined with learning goal setting, 2) HCD activities and 3) game design activities.

[5] Excerpt from the historical mapping session with the project group, transcribed and translated by author.

Concept Generation and Learning Goal Setting

In previous work in this study [2] it has been discovered that the learning goal setting and game concept generation should go hand in hand in GBLE development. It has also been discovered that GBLE projects benefit from iterative approach to game concept generation [2]. The outcome of these activities has also been experienced to be unpredictable as evidenced for example by the unsuccessful termination of the first requirements phase of the Peatland Game project [2].

The notion that these activities are closely knit indicates that the defined process control model, where goals are first set and then a process is planned to meet the goals would be unfeasible for these activities. This is because the both activities are in a way a prerequisite for each other; learning goals cannot be set first and then a game concept created to meet these goals, as the choice of feasible game concepts restricts the attainable learning goals and vice versa.

This is also a matter of planning the resources for these activities as the uncertainty involved would seem to limit the possibility of estimating the resources needed to carry out these activities in a way that enables the desired quality for the work products.

These findings indicate that empirical process control would be a better match for these activities. This would allow the project team to take control of the work being done. The sprint structure of work would allow the activities to be iterated until the work products have been validated to have sufficient quality or the whole project could be terminated after any sprint if the activity would be deemed as unfeasible.

HCD Activities

In the Gameli V.1 project there was a perceived difficulty into fitting HCD activities within a project that used defined process control model. This was perceived to be more because the project planning did not take into account these HCD methods than the inherent incapability to control these processes using defined process control.

In Gameli V.2 and the SR Game projects the incorporation of HCD activities to defined process control was considered more successful, although in Gameli V.2 project the team perceived potential risks related to the combination.

These risks, namely the change in scope and requirements resulting from HCD workshops can be considered as a real one in any HCD project. This comes from the inherent iterative manner of HCD process where user designers are used throughout the project but only in specific points to provide information on requirements and to evaluate work done up to that point.

This kind of work structure would be supported inherently by empirical process control which in itself has the same kind of feedback pattern. For example Scrum every sprint could have one major HCD process iteration where sprint planning and review would be coupled with HCD requirements elicitation and HCD evaluation, respectively.

Game Design Activities

Game design is described as a second-level design activity where the designer designs the rules of the system in order to make the interaction of the player and the game to occur in a specific way [16]. The result of this is that game design can be evaluated only with an end product a player can interact with Ű- a prototype of the game.

The results of this study indicate that this is true to the design of GBLEs as well. Particularly the Peatland Adventure project was reported to benefit from iterative game design, namely producing concurrent versions of the game along the design process. The study of social responsibility game project also reported similar findings.

The iterative game design process demands empirical process control as the iterative game design activity cannot be estimated accurately beforehand. On the other hand, if the project is using traditional document-based game design process the quality of game design cannot be accurately evaluated before the first version of the game is ready.

Although experienced game designers could possible lessen this risk somewhat, it must be concluded that any GBLE development project with a considerable amount of uncertainty related to the game design would benefit from empirical process control related to the game design activities.

The uncertainty can be a result of not having enough results of a certain game type supporting the type of learning intended or not having enough information about the quality of gameplay experience being designed. Projects working with game genres with known advantages of supporting learning in the desired field and a tried and tested game design could therefore be designed with the traditional document-driven approach.

4.3 Process Control Selection in GBLE Projects

This chapter summarizes the findings done in the previous two result chapters regarding experiences on the process control model selection in GBLE projects.

In the case projects risks and problems were encountered with the defined process control model:

- Results of GBLE concept creation can force changes in the project planning,
- risk of feature creep caused by HCD workshop results,
- planning with incomplete goals and requirements leads to problems and
- GBLE tasks that are not explicitly defined are a risk to schedule and resource planning.

These problems were highlighted in projects with much uncertainty in terms of goals and game design solutions.

Many of the key activities of GBLE development were found to be inexplicitly defined or subject to much uncertainty relating to the quality of their end products and the resources needed to complete them.

These finds indicate that, particularly when working with GBLE projects with uncertain elements, an empirical process control model would be preferable to a defined one. In the following chapter we discuss how to adapt the Scrum process model, used as an example of empirical process control to different facets of GBLE development.

4.4 Adapting the Scrum Process to GBLE

Guidelines for adapting the Scrum process, used in this study as an example for empirical process control, to aspects of GBLE development are presented in this chapter. The guidelines are drawn from the literary review of the Scrum process and the experiences in the game projects on carrying out the key activities of GBLE development.

Adapting Scrum to GBLE Concept Creation

This compound activity consists of two sub-activities each producing their own work product. The sub-activities are learning goal definition and GBLE's game concept creation and the corresponding work products are learning goal definition (or requirements) document and game concept document.

As the in Scrum every sprint should produce a shippable delivery of the product and design documentation cannot be considered to be same as the product, adapting this activity to Scrum is not a straight-forward process.

One possible solution would be to add a third sub-activity, producing a prototype of the game concept, to this activity. The prototype would act as the shippable version of the product in Sprints and could be evaluated in Sprint reviews. This would also work towards more valuable evaluation of the other work products of the activity as prototype would offer more evidence to the validity of the learning goals and the ability of the game concept to fulfill them.

This would indicate that in order to keep the process agile the prototype building activity would have to be light, meaning a prototype of low fidelity and therefore also of technology.

Adapting Scrum to HCD

In human-centered design process the three main tasks of the user designer participants are setting the requirements, generating design solutions and evaluating designs and end products [5].

In a project using Scrum these tasks would fit the overall process in the following way: generating requirements would mean submitting items in the project backlog as well as informing the product owner of priority changes of existing items in the project backlog. The design solutions would be created in collaboration with the development team during sprints and the evaluation of end products would be done to inform the sprint review meeting or as part of the sprint review itself.

This indicates that the product owner role should be assigned to the person also responsible for the HCD process and the workshops. Workshops would be arranged at the end of each sprint to review work done and to inform the product owner of backlog changes needed. Additional workshops would be called during sprints to support the design of solutions as needed.

Adapting Scrum to Game Design

Adapting Scrum to game design would mean changing the game design process into a more empirical one and one focusing more on making versions or prototypes of the game along the design process.

This kind of process for game design is described by Eric

Zimmerman in the article Iterative Game Design [16]. Iterative game design is based on a cyclic process in which the production of gameplay prototypes, analysis and refining the prototype take turns [16].

Iterative game design process could be used also in GBLE projects using Scrum. The analysis of prototypes would effectively inform/or take place in the Sprint review meetings and the Sprints itself would be centered on producing and refining the game prototypes.

5. CONCLUSIONS

The aim of this study was to explore the alternatives to process control in GBLE development and provide guidelines of planning and carrying out process control in GBLE projects. The research questions were: 1) What kind of process control model is most advantageous and feasible to the development of GBLE (both in general and especially in those projects concentrating on game design and human-centered design) and 2) what kind of adaptation of that process control model would lead to a high-quality development process in this domain. The research was conducted according to the principles of action research and developmental research. Four case projects were analyzed and their experiences were assessed. In earlier research ([2], [5]) key activities for GBLE development were identified.

In examining the experiences of the case projects regarding process control models and the key activities of GBLE development there were found multiple factors supporting the advantages of using empirical process control model in GBLE development. Problems were noted with activity planning and scheduling in the case projects using a defined process control model. It was found that there were a large number of important activities that were not explicitly defined and thus could not be accurately estimated or planned. The key activities were also found to contain a lot of uncertainty related to their completion. These findings support the notion that utilizing an empirical process control model such as Scrum would be preferable in the development of GBLE.

Scrum was introduced for software development and it must be adapted for game development [3]. The study presents guidelines for adapting Scrum to the learning-goal focused game concept creation in GBLE development, human-centered design and game design activities. These include guidelines of using activities defined in earlier research for development, assessment and control functions within the Scrum framework.

The main results of this study are the evaluation of the two main process control models for the project management of game-based learning environment development as well as guidelines for adapting Scrum for the development of game-based learning environments. These results are based on experiences in the four case projects and a closer examination of key activities in GBLE development identified by earlier research. These results provide a good foundation for future development projects in the field. It will also serve further studies into the process of developing game-based learning environments.

6. REFERENCES

[1] B. Chamberlin. Creating entertaining games with educational content. case studies of user experiences with the childrenŠs website. Master's thesis, University of Virginia, 2003.

[2] A. Kirjavainen. Modeling the game-based learning environment development process. Manuscript submitted for publication. University of Jyväskylä, 2009.

[3] K. Lahti. Adapting to and assessing scrum in game development. Master's thesis, Swedish School of Economics and Business Administration, Helsinki, Finland, 2008.

[4] R. Mcguire. Paper burns: Game design with agile methodologies. *Gamasutra The Art & Business of Making Games*, June 2006.

[5] T. Nousiainen. *Children's Involvement in the Design of Game-Based Learning Environments*. PhD thesis, University of Jyväskylä, 2008.

[6] J. Novak. *Game Development Essentials: An Introduction*. Thomson Delmar Learning, Clifton Park, New York, 2005.

[7] B. A. Ogunnaike and W. H. Ray. *Process dynamics, modeling, and control*. Oxford U.P., New York, 1994.

[8] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still. The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13:303–337, 2008.

[9] Z. Racheva, M. Daneva, and L. Buglione. Complementing measurements and real options concepts to support interiteration decision-making in agile projects. In *34th Euromicro Conference Proceedings*. IEEE Computer Society, 2008.

[10] L. Rising and N. S. Janoff. The scrum software development process for small teams. *IEEE Software*, July/August 2000.

[11] K. Schwaber and M. Beedle. *Agile Software Development with Scrum*. Prentice-Hall, Upper Saddle River, New Jersey, 2001.

[12] I. Sommerville. *Software engineering*. Addison-Wesley, Harlow, 2001.

[13] M. Stubbs and J. Pal. The development, design and delivery of a retail simulation. *British Journal of Educational Technology*, 34(5):651–661, 2003.

[14] L. Tripp, A. Alan, J. Moore, P. Bourque, and R. Dupuis, editors. *SWEBOK - Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, Los Alamitos, California, 2004.

[15] J. van den Akker. *Design approaches and tools in education and training*, chapter Principles and Methods of Development Research. Kluwer Academic, 1999.

[16] E. Zimmerman. *Design Research: Methods and Perspectives*, chapter Play as Research: The Iterative Design Process. MIT Press, Cambridge, Massachusetts, 2003.