



## **Practical Experience: Adopt Agile Methodology Combined With Kanban For Virtual Reality Development**

Bachelor of Science Thesis in Software Engineering and Management

Bin Han  
Jianfeng Xie

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
Goteborg, Sweden, May 2011



The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

## Practical Experience: Adopt Agile Methodology combined with Kanban For Virtual Reality Development

Bin Han  
Jianfeng Xie

© Bin Han, May 2011.  
© Jianfeng Xie, May 2011

Examiner: Helena Holmström Olsson

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
SE-412 96 Göteborg  
Sweden  
Telephone: + 46 (0)31-772 1000

Department of Computer Science and Engineering  
Göteborg, Sweden, May 2011



Bin Han

Dept. Computer Science and Engineering  
University of Gothenburg  
Gothenburg, Sweden  
Email: bin-han@hotmail.com

Jianfeng Xie

Dept. Computer Science and Engineering  
University of Gothenburg  
Gothenburg, Sweden  
Email: kevin\_xiaoxie@hotmail.com

## Abstract

*Development methodology is a part of project management and always plays an important role in software development. In recent years, some new software development methodologies are showing up, like Agile and Lean Software Development. However, as Kniberg [8] stated in his report, there is not a methodology that is appropriate to all software development. Chalmers SAFER Simulation Lab has asked a small team to improve the virtual reality of their driving simulator. Since the previous experience and study of development methodology for virtual reality development is nearly blank. A specific software development methodology is urgently needed to improve project process. Therefore, the purpose of this paper is to develop a specific development methodology that assists this team in managing the project well. To achieve this objective, extensive literature review of development methodology and software process improvement were conducted. The literature review clarified that there are several Agile methodologies that fulfill the specific requirements and constraints of this project in many aspects, but the disadvantages are still obvious. Hence, Kanban as an improvement approach has been applied to overcome these disadvantages. The new software development methodology is called Extremeban, which is a combination of Agile features and Kanban. Extremeban has been applied in the project as experiment. This paper brings together a complete set of evaluating agile methodologies, introducing new software development methodology (Extremeban).*

**Keywords:** *Software Process Improvement, Virtual Reality Development, Agile Software Development, Kanban,*

## 1. Introduction

Driving simulators are being increasingly used for research, training and business in recent years. Some vehicle manufacturers operate driving simulators to validate the quality of their products. Driving simulators have also been used by many universities or institutes to observe drivers' behavior under conditions which are impossible or illegal in real world. There are many development projects related to driving simulators nowadays. These projects aim to fulfill the growing needs from customers, and for further academic or technique research.

Chalmers University of Technology founded SAFER Simulation Lab in 2006. SAFER hammers away at excellent multi-disciplinary research and collaboration to eliminate fatalities and serious injuries, making Swedish society, academy and industry a world leader in vehicle and traffic safety. SAFER bought a driving simulator STISIM Drive 2000 a couple of years ago. This driving simulator was produced by an USA company, so most of the default objects, e.g. terrain, building, car, are based on USA environments. SAFER wants to develop a Swedish virtual reality environment for this driving simulator. Four undergraduates from the University of Gothenburg are involved in this development project. An appropriate development methodology is important for any software development project. Software development methodologies ensure projects to deliver right products on time and keep cost within budget. However software development projects are different in many aspects, e.g. development scale, project properties, developers, specific requirements, and constraints. It is obvious that there is no development methodology that can be used in all kinds of projects. In order to complete SAFER project smoothly, an appropriate development methodology is needed. The purpose of



this paper is to analyze existing agile development methodologies and develop a new software development methodology (Extremeban) that is specific for the SAFER project. This new methodology has been applied to the SAFER project.

An extensive literature review will be used for clarifying the theoretical concepts of agile methodologies, software process improvement and virtual reality development. We will first analyze major challenges for the SAFER project. After that, according to previous research and experience, a number of agile development methodologies will be introduced and analyzed. Then, by evaluating development challenges and characteristics of existing methodologies, we will illustrate our solution: an improved methodology combined with new technique. This new software development methodology (Extremeban) has been applied in the SAFER project as an experiment. Finally, some practical experience or tips will be delivered after applying Extremeban. Our main research will be introduced separately as follow:

### *Study/Analyze Development Methodology*

An appropriate development methodology is important for any software project. In section 2, the paper will introduce concepts of agile development methodologies and lean development approach, including their properties, benefits, and shortcomings. Depending on the result of comparing existing agile methodologies with constraints of the SAFER project, the most suitable methodology (Extreme Programming) has been selected as the base for new methodology.

### *Challenges of SAFER Project*

Comparing with other usual software development projects, there are a lot of specific constraints in the SAFER project. In section 4, based on data gained from our meeting with SAFER, we will identify specific constraints and challenges, which arose during the development. These constraints are important as they will influence the choice of software development methodology for this project.

### *Develop a New Methodology: Extremeban*

Though many properties of Extreme Programming are suitable for the SAFER project, there are still some challenges if adopted directly. Extreme Programming needs to be modified and improved to overcome the challenges to meet specific requirements of the SAFER project. In section 5, the paper will describe one approach of Lean Software Development (Kanban) to overcome these challenges. A new software development methodology, Extremeban, will be defined based on Extreme Programming and Kanban.

### *Extremeban Experience*

Extremeban was applied in the SAFER project in practice. Section 5 will also present the results issued from the project

## **2. Background**

In this section, some general information of the SAFER project (e.g. organization, objective, motivation, etc.) will be introduced. After that, a literature study will be used to introduce the concepts of software development methodology, and state its importance in development project. Specially, two agile software development methodologies (Extreme Programming and Scrum) and Lean development approach (Kanban) will be focused on.

### **2.1 The SAFER Project**

SAFER Simulation Lab bought a driving simulator STISIM Drive 2000 a few years ago. This driving simulator was produced by a company from US, so most of the default objects, e.g. terrain, building, car, are based on USA environment. To improve its localization, SAFER wanted a Swedish virtual reality environment instead. Hence, they asked four undergraduates from the University of Gothenburg to get involved in the SAFER project to develop a Swedish virtual reality environment for this driving simulator.

Virtual reality is a term that applied computers to simulate a physical presence representing places in the real world, as well as imaginary worlds. To build a realistic environment in computer, the first step is to



develop a 3D model which is a mathematical representation of any three-dimensional surface of object (either inanimate or living) via specialized software. What's more, some short scenarios are written to import these 3D models, as well as predefine the order of actives.

In the SAFER project, in order to help experimenting user behavior in a more specific scene, the new virtual reality environment should include characteristics tailored the Swedish traffic system, including traffic signs, transportation, road construction, etc. To complete the project objectives, the project team must firstly construct new 3D models (e.g. traffic signs, buildings, transportation, etc.) to replace of default ones. 3DS MAX is the software tool used to develop these models. Secondly, the project team must write scenario event files in SDL (Scenario Definition Language) to simulate the locations and activities of new developed objects. SDL is the built-in scenario programming language of STISIM Drive 2000. It defines virtual environment and when/where specific event happens, just like a short scenario in a film.

## 2.2 Literature Study

This section will introduce the definition of software development methodology, several agile software development methodologies and one lean software approach. Data is collected from relevant papers, reports and books. The objective is to study their characteristics and properties to figure out what kind of project each is suitable for.

### 2.2.1 Software Development Methodology

A software development methodology is a set of activities that lead to the production of a software product [7]. Every software development methodology framework acts as a basis for applying specific approaches to develop and maintain software. The frameworks of these methodologies are used to structure, plan, and control software process in projects. The objective of using software development methodology is to improve productivity and quality of a project. "Without project management, software projects can easily be delivered late or over budget. Because of the need for judgment and creativity,

attempts to automate software development methodology have met with limited success" [7]. Several software development methodology approaches have been used since the origin of information technology [24]:

- Waterfall: a linear framework
- Prototyping: an iterative framework
- Incremental: a combined linear-iterative framework
- Spiral: a combined linear-iterative framework
- Rapid application development (RAD): an iterative framework
- Extreme Programming

Software development methodologies have an impact on the success of a project. They are important to all projects, but the fact is no single software development methodology framework is suitable for all kinds of projects. Each of the available methodology frameworks are best suited to specific kinds of projects, based on various technical, organizational, project and team considerations [24]. Kniberg [8] had a similar opinion about this: there is not a methodology that is appropriate to all software development. In [6], Summerville mentions that for instance, for critical systems, a very structured development process is required; for business systems with rapidly changing requirements, a flexible, agile process is likely to be more effective.

### 2.2.2 Agile Software Development

Agile software development (Scrum, Extreme Programming, etc.) and traditional Plan-Driven software development (Waterfall, Spiral etc.) are two major methodology types for software development. Table 1 lists characteristics of Agile methods and Traditional Plan-Driven Methods.

Kumar [12] stated that centralized decision making and traditional software practices are two root causes of software failure. According to analyze the characteristics of Plan-Driven Methods in Table 1, we found they are not appropriate for this SAFER project in the following aspects:



1. Plan-Driven Methods require developers with adequate skills, but many techniques (e.g. SDL and 3D modeling) for the SAFER project were new to the developers
2. The requirements of the project need to be knowable early and largely stable. However, the SAFER did not have enough knowledge and experience on virtual reality development. It resulted in a difficulty to predefine the stable requirements.
3. Plan-Driven Methods appropriate for larger teams and products, but the development team is small.

On the contrary, Agile Methods are more appropriate in these aspects (e.g. embrace rapid change, smaller teams/products and collaborative developers). Therefore, we will focus on Agile Methods in this paper. Extreme Programming (XP) and Scrum will be introduced in this section. Additionally, in order to face several root causes that Kumar [12] listed, e.g. frequently/rapidly changing customer requirement and rigid project scope management, one of the Lean software development approaches (Kanban) will also be introduced.

### 2.2.2.1 Extreme Programming

“Extreme Programming (XP) is a popular methodology of software development; its goals are to improve productivity, flexibility, informality and limited use of technology outside of programming” [5].

As one of Agile Methodology, XP contents a number of short life cycles with frequent test and “release”. Every cycle sets objectives which include a subset of requirements form complicated or larger set, and the cycle is called sprint. XP relies on four values; simplicity, communication, testing and courage, and each practice enhance the others [5]. In detail simplicity represents a way of everything, e.g. design, test and research starts from the simplest task, and keeps the items in the simple condition all the time. Communication encourages face to face talking and exchanging ideas among developers, manager and etc. A good example of encouraging communication is pair programming, it means that two people working together with the same piece of code. This way has many advantages, e.g. the quality of the code is higher [6], the skills of the members of the teams develop more evenly, and the success of the project does not rely on a super-programmer but on teamwork [5]. Another character of XP is test being done during the entire project, but more importantly, the tests drives implementation. Programmers should complete to write functional testing for each piece of code before starting to write these codes. This approach intends to make programmer think about what potential problem they will meet and conditions in which the code will fail. Courage means that members of team must have to address problem with self-confidence. If some work completely goes wrong, it is important not to hesitate to throw it away; start over again instead of trying to fix or recover it [5].

Area	Agile Methods	Plan-Driven Methods
Developers	Agile, knowledgeable, collocated, and collaborative	Plan-oriented; adequate skills; access to external knowledge
Customers	Dedicated, knowledgeable, collocated, collaborative, representative, and empowered	Access to knowledgeable collaborative, representative, and empowered customers
Requirements	Largely emergent, rapid change	Knowable early, largely stable
Architecture	Designed for current requirements	Designed for current and foreseeable requirements
Refactoring	Inexpensive	Expensive
Size	Smaller teams and products	Larger teams and products
Primary objective	Rapid value	High assurance

Table 1. Home grounds for agile and plan-driven methods [10]

XP involves 12 practices and grouped in 4 areas, they are: planning game, small releases, metaphor, simple design, test, refactoring, pair programming, continuous integration, collective ownership, on-site customer, coding standards and 40-hour week.

### 2.2.2.2 Scrum

Hiroataka Takeuchi and Ikujiro Nonaka described a new approach to commercial product development that would increase speed and flexibility, based on case studies from manufacturing firms in the automotive, computer, photocopier, and printer industries [2]. Scrum is an iterative, incremental framework for project management often seen in Agile software development.

Scrum advocates the use of small teams - no more than 10 team members [3]. Besides of these team members, Scrum contains a Scrum Master to maintain the processes and Product Owner to represent the stakeholders and the business. The entire development process is divided into several sprints, and each sprint lasts one to four weeks (shown in Figure 1). In the initial planning phase of Scrum, the project team must define architect and features as backlog of the project, which is a prioritized set of high level requirements of work to be done. Thus, the entire team must have a single target and the priorities must be clear [3]. Scrum allows requirements, features or other modifications change during the development process. This is one benefit of splitting process in several sprints. However, if requirements for one sprint were frozen, no change would be allowed until the end of the sprint. After each sprint, the development team report and show all new developed features to all stakeholders. Then a new plan for next sprint will be made through communication with acquirers.

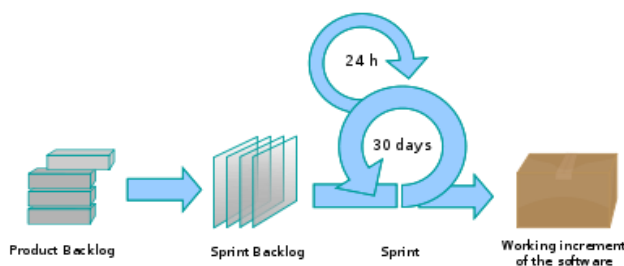


Figure 1: Scrum Process

Daily scrum meeting in the morning is one well-known characteristic of Scrum. The meeting is time-boxed from 15 to 30 minutes and led by Scrum Master. During the meeting, each team member answers three questions [4]: What have you done since yesterday? What are you planning to do today? Do you have any problems that would prevent you from accomplishing your goal? This helps team members to focus the effort on backlog, keep everyone informed of team progress and obstacles, resolve obstacles and tracking progress.

### 2.2.3 Lean Software Development

By analyzing these Agile methodologies, we found these methodologies are still not agile enough as we expected. Each of them more or less contains rigid rules and limitations. “The problems of the software development planet are responsible for most of the project failures that force managements worldwide to put more rigid processes in place to ensure compliance” [12]. Lean Software Development is a translation of Lean manufacturing and Lean IT principles and practices to the software development domain. Lean Software Development originated in the book written by Mary and Tom Poppendieck [11]. Mary and Tom Poppendieck summarized and listed the following seven Lean principles in their book [11]:

1. Eliminate waste
2. Amplify learning
3. Decide as late as possible
4. Deliver as fast as possible
5. Empower the team
6. Build integrity in
7. See the whole

Lean Software Development can be considered as a new development method that tries to identify and eradicate all problems and “disabilities” of old methodologies [13]. It helps software organizations to optimize development processes and methods, improve efficiency and product quality. “By using Lean Production Manufacturing principles not only quality concerns and other issues can be resolved, but also a continuous improvement cycle can be built in the process” [12].

### 2.2.3.1 Kanban

Kanban is a lean approach to agile software which was pioneered by David Anderson as a more direct implementation of Lean Thinking and Theory of Constraints to software development in 2004 [9]. In 1950s, Kanban was a logic used by Toyota to tie together the visual and physical signaling system. Even though Kanban has been used for over a half century in Lean Production, it is still a new one in software development area.

Kniberg introduced the way Kanban works in his report [8]:

- Visualize the work flow
  - Split the work into pieces, write each item on a card and put on the wall.
  - Use named columns to illustrate where each item is in the work flow.
- Limit WIP (work in progress)
  - Assign explicit limits to how many items may be in progress at each work flow state.
- Measure the lead time (average time to complete one item, sometimes called “cycle time”);
  - Optimize the process to make lead time as small and predictable as possible.

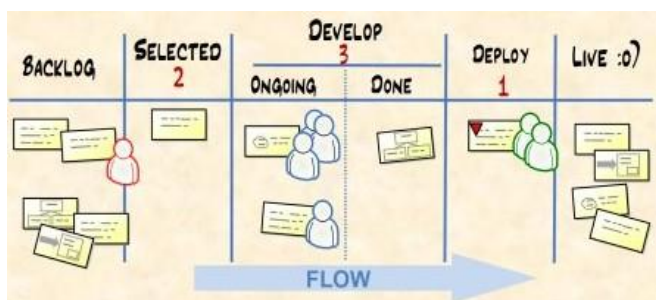


Figure 2: An example of Kanban board

Figure 2 is a Kanban board example. The work flow of each task is displayed in separate columns clearly. In this example, the work flow includes five steps: Backlog, Selected, Develop (Ongoing, Done), Deploy, and Live. Each paper on the board is one task, including specification, requirements, developers' names, and deadline (circle time) etc. These data on task paper are editable at any time, which provides a flexible management. The numbers under some steps are WIP. This number limits the quantity of tasks

which can be executed at the certain step at the same time. WIP forced developers to focus on several most important tasks. The work flow is from left to right. Once the developer finished one step for one task, it should be moved to next step. Sometimes the work flow can be reversed if any task needs to rework.

Unlike Scrum dividing entire project into several sprints and each sprint last a certain amount of time, Kanban provides a way to do agile software development without necessarily having to use time-boxed fixed-commitment iterations. The work flow of Kanban is continuous. Kanban board clearly shows that the entire project status in real time like what has been done, what need to be done, and bottlenecks if any task stays in ongoing status for a long time.

## 3. Research Approach

In this section, the research approach we used will be introduced, including research setting, research process and data collection. Besides, we will state limitations of the research approach and what we have done to minimize the negative effects.

### 3.1 Research setting

The research is based on virtual reality development in the SAFER project. In the SAFER project, they had set up a driving simulator, which was bought from US Company STI. Four Chalmers students helped carrying out experiments in the driving simulator as technical assistants. However, the problem was that the default objects in virtual reality were all based on US buildings, roads, traffic signs, etc. It would not meet experiment requirements in Sweden. Therefore, SAFER asked the team to help develop a Swedish virtual reality environment to replace the default one. Since this virtual reality development differs from other usual software development, there are several challenges during development. This research aims to develop a specific development methodology as a solution to overcome these challenges. Furthermore, this new methodology was applied in virtual reality development of the SAFER project as an experiment.





### 3.2 Research process

The research consists of three main components: challenges of the SAFER project, development methodologies analysis/study and a new methodology development. The first step is to acquire more knowledge about virtual reality development in the SAFER project. We had a meeting with SAFER Simulation Lab staff to get a clear understanding on their purpose and requirements for this project. In this meeting, the project acquirer introduced their organization and illustrated requirements of this project. We also visited VCC HMI Lab (Volvo Car Company's Human Interaction Laboratory) to gain some practical experience on using and maintaining a driving simulator. By analyzing this information together, we identified major challenges of virtual reality development in the SAFER project.

Based on several identified challenges, we evaluated these agile methodologies and lean development approach to find out what features are appropriate to virtual reality development in the SAFER project. The solution we recommended is to combine the agile methodology and lean development approach to develop a new improved methodology.

### 3.3 Data collection

Data collection of this research was done in two ways. The first way was to obtain secondary data from the previous papers related to software development methodology and technology. We searched for relevant papers using search engines (e.g. Google scholar, Springer Link, IEEE Xplore and Elsevier.). The second way was to get primary data from our meetings with developers and acquirers in the SAFER project. Also, we contacted with a maintainer in VCC HMI Lab to get some experience on how to use and maintain a driving simulator.

### 3.4 Limitation

There are several development methodologies which may be appropriate to the SAFER project. However, with the resource limitation, we were not able to build several teams to test each of these development methodologies and compare results to evaluate which one was more appropriate than the others. So the

development methodology that we used was based on our experience and previous research. These development methodologies are process tools. "No tool is complete, no tool is perfect" [8]. Different projects have diverse requirements, backgrounds, organization and schedule. Even though our development methodology was appropriate to the SAFER project; we could not guarantee this could be used in other projects. Moreover, due to time limitation, we could not consider each perspective of the development methodology. Hence, a few issues have been neglected.

## 4. Results

This section will present findings from the meeting with SAFER. Data from our meeting with developers and acquirers in the SAFER project focused on what challenges were present during virtual reality development.

### 4.1 Major challenges for virtual reality development in the SAFER project

As mentioned in Section 2.2.2, each software development methodology is only suitable for specific projects. In order to select a suitable methodology, there is a need to understand specific properties and challenges. Several major challenges of the SAFER project will be identified in this section. These challenges are categorized in four aspects [24]: technical, organizational, project, and team challenge.

#### 4.1.1 Technical challenge

First of all, STISIM Drive provided Open Module and SDL as tools to set values, modify events, and add new objects into the virtual reality of STISIM Drive 2000. Open Module and SDL consist of specific programming languages, contents and arguments. Compared with other language, SDL is a light-weighted language for run-time interaction. Furthermore, there is no architecture and unit test in SDL programming. Hence, the team members needed time to study and accommodate to this new programming. Secondly, as the acquirer of this project, SAFER Simulation Lab, mainly focused on HCI (Human Computer Interaction) and HMI (Human



Machine Interaction) research in driving area, lacked knowledge about software engineering and management.

Additionally, since the driving simulator had been set up and operated only for three months, SAFER Simulation Lab had not used STISIM Drive 2000 in-depth yet. Some functionalities of this simulator were still a blind area for them. Thus backup and technical supports from SAFER Simulation Lab to the project team were limited.

#### 4.1.2 Organizational challenge

Most of the development methodologies require a lot of resources, which are not available in small firms or organizations [1]. For instance, waterfall is more suitable for large scale projects, because it requires high-level of human resource. However, the development team in the SAFER project was composed of four members. It was necessary to take team size into consideration while selecting development methodology.

#### 4.1.3 Project challenge

Stable development requirements are seen as a pre-requisite for starting a software development project. However, as acquirers of the SAFER project, the researchers did not have enough knowledge and experience on virtual reality development, it was very difficult for them to predefine development requirements correctly and completely in the beginning of the project. Hence, requirement changing is an unavoidable challenge during development.

The duration of the SAFER project was three months. The development team had to complete design and implementation within limited time. Hence, the software development methodology for this project should be highly efficient to ensure that the team can complete all tasks in time.

#### 4.1.4 Team challenge

Before this project started, none of the team members had any professional experience or knowledge in driving simulator development and SDL programming. 3D modeling was also a new technology for them.

Hence, the team members were junior developers in this field. However, some development methodologies require developers with high-level design and programming ability, which the team members could not be able to reach. Because of this factor, there is a need to consider which development methodology would be helpful to improve group productivity.

## 5. Discussion

This section will give feedback for these findings in Section 4. Two agile methodologies and one lean development approach (as introduced in Section 2.2.2) will be evaluated to find out what features of them are appropriate to virtual reality development in the SAFER project. After that, there will be a sufficient description on how to combine methodologies together to develop a new improved methodology for the SAFER project. Furthermore, the result of applying Extremban in the SAFER project will also be brought out.

### 5.1 Agile methodologies evaluation

As described in Section 2.2.2, there are two Agile development methodologies, XP and Scrum as candidates for virtual reality development in the SAFER project. This section will evaluate Scrum and XP separately by assuming that each is applied in this project. Therefore, it will present what features have a positive or negative effect on this project.

#### 5.1.1 Extreme Programming

In Section 2.2.2.1, there is a brief introduction of Extreme Programming. Currently we will discuss the advantages and disadvantages of applying Extremban in the SAFER project. Firstly, compared with typical project meeting in which attendees do not contribute, but hear outcomes, a stand up meeting in XP is used to communicate problems, solutions, and promote team focus. Everyone stands up in a circle to avoid long discussions [20]. In the SAFER project, when the team intended to set up a meeting, the attendees were people who would be needed or contribute to the discussion. It is more productive that the short meeting with limited attendees replaces long meeting with everyone. This way would help to save much resource and time for the



team, and is possible to decrease meeting absence. Secondly, since it was reported that “XP is also easy to learn, reduces overhead, and provides greater productivity. However, it is limited in terms of team size, and its inability to scale. For instance, it is best used with small to medium sized teams of no more than 12 people highly skilled and motivated individuals” [14]. Hence, XP was very suitable for this four-person team, but it brought an issue at the same. The team members’ skills in SDL programming and 3D modeling were not as high as XP required.

## 5.1.2 Scrum

There are several characteristics of Scrum which are appropriate to the SAFER project, while some are not. Firstly, breaking down a large set of requirements into smaller pieces is helpful to advance the SAFER project process. The team can divide members into two groups: Scenario/Open Module, and 3D modeling. Hence, they will work in parallel in different components to decrease the development duration, which is one of the challenges we mentioned in Section 4.1. Secondly, because daily Scrums meetings improve communications, eliminate other meetings, identify and remove impediments to development, highlight and promote quick decision-making, and improve everyone’s level of project knowledge” [23]. Daily Scrum meetings can help the team to overcome communication barrier, track progress, and draw up a daily plan.

In Rising and Janoff’s report [3], they demonstrated an experience report from diverse software development teams which used Scrum. The team leader found it challenging to use Scrum: we needed someone who could facilitate a tight meeting, keep everyone on track and solve problems on the fly. Thus, having a capable Scrum Master is very significant to guide a team. However, all team members in the SAFER project are junior developers. The Scrum Master challenge for other development teams is also a trouble for them. Besides, the second inappropriate practice of Scrum for this project is time-boxed sprints. Scrum splits project in several time-boxed sprints, and nothing can be changed until one sprint is finished. This impact on rapidly changing requirement challenge for this project.

Acquirers will have new idea or change requirement at any time and the team maybe face unexpected technical issues during development because of technical challenges (as introduced in section 4.1). Therefore, the team needs a more flexible method which allows changing project plan, schedule and tasks anytime.

## 5.2 Extremeban: modified methodology with Kanban

As discussed above, Scrum and XP are neither complete nor perfect for our project. Herink and Skarin stated [16]:

“Don’t limit yourself to only one tool. Mix and match the tools as you need! I can hardly imagine a successful Scrum team that doesn’t include most elements of XP. For example, many Kanban teams use daily stand up meetings (a Scrum practice). Some Scrum teams write some of their backlog items as Use Cases (a RUP practice) or limit their queue sizes (a Kanban practice). Whatever works for you?” We suggest that methodologies combination is the best approach to deal with these challenges in the SAFER project. The modified methodology is based on XP methodology and improved with Kanban. Kanban encourages incremental evolution of existing processes and evolution that is generally aligned with Agile and Lean values [8]. In detail, the modified methodology (Extremeban) contains main characteristics of XP, and Kanban as management tool is attached to development in the SAFER project.

Figure 3 illustrates the overall structure and dependencies between different practices in Extremeban. In the rest of sections, we will describe the details of them one by one.

### 5.2.1 Training

In order to make the framework successful, all of the people involved in software development must have a good knowledge in development and they must be trained [1]. The Extremeban is modified based on XP and Kanban. It is a new software development methodology for all team members. The best way to learn this development methodology is to provide a training phase for the entire team, e.g. introduce the modified methodology, what is it and how to use it.

Furthermore, as we have stated in Section 4.1, if the team members were junior developers who got unfamiliar with technologies, the training phase extended to study development tools and new programming languages. Hence, the training time needed to be estimated by all team members. It would impact the release plan.

### 5.2.2 User stories

XP organization said that “User stories serve the same purpose as use cases but are not the same. They are used to create time estimates for the release” [21]. User stories are used to replace of traditional large requirement specification. In order to prepare for release planning, a customer expresses or writes his/her needs for what system to do using index card or “user stories”. These stories are not only limited to user interface, but also provide enough details for what system to do and strategies to avoid technical issues (e.g. algorithms and data layout). Besides, they help estimating implement time for each task, and driving acceptance test. These are which acceptance should be done to validate whether user stories is implemented correctly.

### 5.2.3 Release planning

A Release planning phase aims to make a release plan, which conducts whole project depending on the iteration plan. In release planning meetings, developers identify all tasks required to meet User stories and estimate deadlines for each User story. From previous

experience, every story needs 1, 2 or 3 weeks to estimate in “development ideal time”. Development ideal time means how long time a User story will cost to implement in code if there is no further requirements or distractions. If development ideal time is longer than 3 weeks, developers need to break down it into smaller piece. If development ideal time is shorter than 1 week, it means that you are too details at level, then combine the story [21].

After collecting user stories, developers and users make a decision together to prioritize subset of stories, which should be completed earlier. Furthermore, the iteration plan is also made, including date of release, duration, tasks, acceptance testing etc. In XP, the iteration and release are frequent. Even requirements are ill defined up front; there is a good chance that they will change when the user sees more of the application. XP embraces changes and allows the users to have continuous participation in the software design [14].

### 5.2.4 Iteration

From this phase, teams start to use Kanban board. By using Kanban, the work flow is directed in a way that allows for minimum completion time for each user story or programming error, and on the other hand ensures each team member is constantly employed [8]. Compared with Agile, Kanban is not time-boxed. Developers can add or modify tasks during development phase. If customers had a new requirement, it is flexible to change task priority, schedule, etc. Hence, iteration helps a team to deal

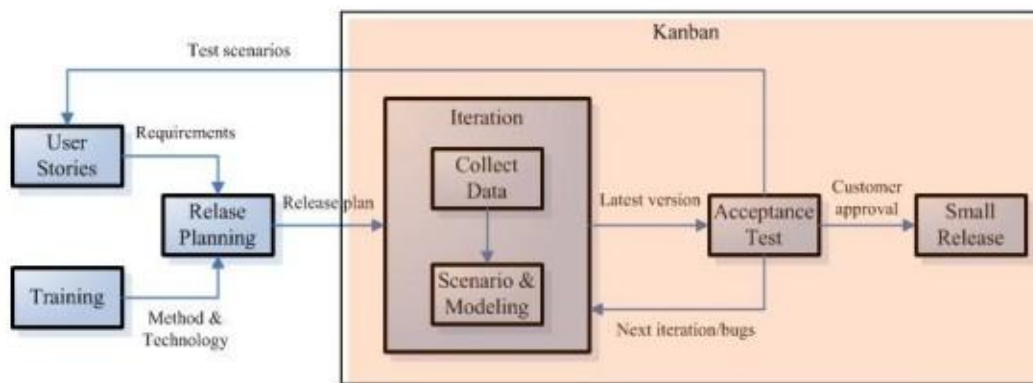


Figure 3: Modified process methodology (Extremeban)

with customers' rapidly changing requirements. On the other hand, Kanban is also a good tool to track development progress. It clearly shows what developers are doing, what need to be done, what need to be tested, or bottlenecks if any task stays in the same place for a long time. Continuous work flow of Kanban ensures each task can go in parallel without wasting of resource, to enhance the team productivity.

In the SAFER project, the Kanban board included five columns: backlog, selected, development, acceptance, and release (as shown in Figure 4). The effect of limiting WIP provides predictability of cycle time and makes deliverables more reliable [16]. WIP (work in progress) limitation was 2 for selected, 3 for development. There was at least one index card for each team in selected column until the end of the project, thus WIP limitation was 2. The Scenario members worked together, but the modeling members built their own objects separately (e.g. one model bus stop, one model building.), therefore WIP limitation for development was 3. There were two sub-columns for development on the Kanban board: on progress, and done. Team members selected 2 index cards from backlog and moved them to selected column, which showed the next task needed to be done. To build models (e.g. building images and location map), the team member needed to collect data. Thus, iterations of this project included two parts: data collection and scenario/ modeling. If the data was prepared and somebody started working on tasks on an index card, it needed to be moved to the development column. Another index card had to be selected from the backlog column to fill the space.



Figure 4: Practical Kanban board

Quality is very important for each software development project. How to guarantee product quality becomes a major issue, especially for junior developers. "Pair Programming increases software quality without impacting time to deliver" [15]. Pair programming is a good solution to increase quality; meanwhile it doesn't impact time limitation of the project. Hence, we suggest teams to apply pair programming in iterations.

### 5.2.5 Acceptance test

"The first step in lean thinking is to understand what "value" is and what activities and resources are absolutely necessary to create the value" [12]. Eliminate waste is one of Lean Manufacturing Principles [11]. As introduced before, acquires of the SAFER project lacked of knowledge on software development. Therefore, requirements for the product were usually unclear and changing all the time. Due to time and manpower limitation, overproduction needed to be avoided. At the same time, the development team was responsible to improve product quality and meet acquires' requirements. Thus regular acceptance test is necessary.

In this project, the acceptance test was a black box test. All developers and relevant stakeholders needed to be present. All index notes from development-done column were moved to acceptance. In the latest version of scenario, all new developed models were imported and executed on the driving simulator. During test process, developers elaborated what had been done in the last period and recorded comments/feedback from stakeholders. If any task on index card needed to be modified, write down the problems on the card and move it to selected column. The developer who was responsible for this task estimated when to fix it himself. The only rule was that the problems found in current acceptance test must be fixed before next acceptance test.

### 5.2.6 Small release

At the end of each iteration, the development team needed to release the latest version of the application to the customer. As James [17] stated that these small releases are incremental production versions of the project's expected final deliverable providing limited



subsets of functionality to the system's users. Each release of additional functionality offers stakeholders an opportunity to use the evolving capabilities of the software and provides high quality feedback, thereby improving the quality of progressive elaboration [17]. Earlier release could help user to realize what was project going on and how was the product, decreasing the following risk: Spend a lot of time delivering things that are valuable to the customer, which strains the relationship with the customer. Try to make the infrastructure cover everything might need, which leads to an overly complex infrastructure [18]. Hence, if a project went wrong it would give an early warning sign of the project in trouble.

### 5.3 Extremeban experience

The SAFER project team applied the Extremeban for virtual reality development. In general, it was appropriate to this project and the result was satisfactory. This section summarizes the results issued from this project.

#### 5.3.1 Daily meeting

Daily meeting in the morning was adopted at the beginning of the project. The development team carried it out for several days, and then troubles came unexpectedly. All team members were bachelor students who were not able to attend meetings every day for different reasons: part time job, exam, courses, and other personal situation. As a result, the task progress could not be reported and tracked all the time, leading the deliver delay. To address this issue, the development team was recommended to set a short meeting with limited attendees instead of a long meeting with everyone.

#### 5.3.2 Team enthusiasm

Since the development team applied a new development methodology, group members needed to learn it beforehand. Developers were required to have enough learning capacity and enthusiasm. On the other hand, there was no role of group manager in the SAFER project; it led to difficult management without decision maker. In order to avoid it, all developers needed to be more motivated to deal with issue they face independently.

#### 5.3.3 Methodologies combination

In this research, the idea to combine XP with Kanban was taking from previous research [16], which discussed the reasons on why to combine Scrum with Kanban. Through our study, we presented methodologies combination is possible in a practical manner.

#### 5.3.4 Kanban Usage

In general, adopting Kanban was a good choice for the SAFER project. It brought a continuous development process and visualized work flow for the team and customers. However, there were also some problems which need to be solved in the future. The capability of entire team needed to be well estimated while setting Work In Process (WIP). If WIP limitation was set higher than the capability of project team, there would always be too many tasks in Backlog, so that team members would have no time to help each other. On the contrary, if reduced WIP and made more team members work on the same task, it would be more efficient. Furthermore, software quality impacts on continuous development. Team members need to do their best to complete a task right at first time to reduce rework. This is also what Lean development recommends.

## 6. Conclusion

This paper set out to seek a development methodology and technique which were appropriate to driving simulator virtual reality development in small team. We specifically focused on developing a new software development methodology (Extremeban) to face different challenges of the SAFER project. Combination of XP and Kanban is a new attempt in software development field. The outcome of this attempt can be guidance to organizations which are interested to adopt Lean software practice, especially Kanban in software development. On the other hand, for that software development teams that face similar challenges like the SAFER project, this research can be seen as a good case. In terms of future research, we are going to validate Extremeban in more software development teams and projects to complete and generalize it.



## **7. Acknowledgement**

We thank Agneta Nilsson, Fang Chen, Helena Holmström Olsson, Tomas Arts, SAFER Simulation Lab and VCC HMI Lab for their support and guidance with providing ideas and feedback during research. Sincerely thanks our academic supervisor Jonas Öberg and Gerardo Schneider for their invaluable support that helps us finish the research.



## Reference

1. Altarawneh, H. & Shiekh, A.E., (2007), *Theoretical Agile Process Framework for Web Application Development in Small Software Firms*, Sixth International Conference on Software Engineering Research, Management and Applications
2. Takeuchi, H. & Nonaka, I., (1986), *The New New Product Development Game*, Harvard Business Review
3. Rising, L. & Janoff, N.S., (2000), *The Scrum Software Development Process for Small Teams*, IEEE Software, vol. 17, pp. 26
4. Schwaber, K., (2004), *Agile Project Management with SCRUM*, Microsoft Press, pp. 135
5. Francisco, M., Mike, H. & Marian, G., (2003), *A Formal Experiment Comparing Extreme Programming with Traditional Software Construction*, Computer Science 2003, pp. 73
6. Williams, L., Kessler, R.R., Cunningham, W. & Jeffries, R., (2000), *Strengthening the case for pair programming*, IEEE Software, vol. 17, pp. 19
7. Sommerville, I., (2006), *Software Engineering 8th Edition*, Addison-Wesley, New York, USA
8. Kniberg, H., (2006), *Kanban vs Scrum - How to make the most of both*, CRISP
9. Kanban (n.d.), Available from: <http://www.crisp.se/kanban>, (Accessed 2 May 2011)
10. Rahimian, V. & Raman, R., (2008), *Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach*, Research Challenges in Information Science 2008, pp. 337
11. Poppendieck, M. & Poppendieck, T., (2003), *Lean Software Development: An Agile Toolkit*, Addison-Wesley Professional
12. Kumar, D.R., (2005), *Lean Software Development*, Project Perfect
13. Bielicki, P., (2009), *Seven Principles of Lean Software Development*, Available from: <http://agilesoftwaredevelopment.com/leanprinciples>, (Accessed 27 April 2011)
14. Bahli, B. & Zeid, E.S.A., (2005), *The role of knowledge creation in adopting extreme programming model: an empirical study*, Information and Communications Technology 2005, pp. 75
15. Wells, D., (2009), *Pair Programming*, Available from: <http://www.extremeprogramming.org/rules/pair.html>, (Accessed 23 April 2011)
16. Kniberg, H. & Skarin, M., (2010), *Kanban and Scrum: making the most of both*, Enterprise Software Development Series
17. Goebel, C.J. III, (2003), *Extreme Programming Practices Used to Facilitate Effective Project Management*, Menlo Innovations
18. Beck K. & Fowler, M., (2001), *Planning Extreme Programming*, Addison-Wesley
19. VCC (Volvo Car Cooperation), (2010), *VCC HMI Laboratory Report*
20. Wells, D., (2009), *Daily Stand Up Meeting*, Available from: <http://www.extremeprogramming.org/rules/standupmeeting.html>, (Accessed 25 April 2011)
21. Well, D., (2009), *User Stories*, Available from: <http://www.extremeprogramming.org/rules/userstories.html>, (Accessed 21 April 2001)
22. Mollet, N., Brayda, L.G., Chellali, R. & Fontaine, J., (2009), *Virtual Environments and Scenario Languages for Advanced Teleoperation of Groups of Real Robots: Real Case Application*, 2009 Second International Conferences on Advances in Computer-Human Interactions
23. Schwaber, K., & Sutherland, J., (2010), *Scrum Basics*, Scrum.org
24. Centers for Medicare & Medicaid Services (CMS) Office of Information Service (2008). *Selecting a development approach*. Webarticle. United States Department of Health and Human Services (HHS). Re-validated: March 27, 2008. Retrieved 27 Oct 2008.