Agile Development of WebVR Applications

Zhixing Shen and Jinyuan Jia

School of Software Engineering, Tongji University, Shanghai, China

***Abstract*—This paper presents a novel "Roll & Raid" WebVR development model to solve the problems of conventional WebVR development practices. Firstly, agile methodology is introduced into the life cycle of developing WebVR. Secondly, some principles are further proposed to guide the entire WebVR development process. We choose some typical WebVR projects to verify its effectiveness. The experimental results show that our proposed "Roll & Raid" model is much advantageous over conventional WebVR development methods with lower cost, higher efficiency, stronger adaptiveness and easier team management.**

***Index Terms*—WebVR, Software Engineering, Agile, Roll and Raid.**

I. INTRODUCTION

Since VRML (Virtual Reality Modeling Language) and X3D (eXtensible 3D) were established as Web3D international standards by ISO in 1997 and 2004 successively [1], more and more Web3D contents emerges on Internet. When traditional webpages are embedded with Web3D contents, users can immerse such Web virtual worlds as campus, shopping mall, tourism spot, museum and et al. Also, you can play a funny online WebVR game without installing any WebVR plug-ins at the first time. Furthermore, you can preview products and even customize them virtually online. WebVR provides a fascinating and “alive” Web3D experience with immerse, interaction and imagination. Currently, huge mainstream WebVR applications have been developed successfully already, Second life is one most representative of them. The emerging Shanghai Web Virtual Expo will become another famous representative soon. Most of the existing WebVR applications have been developed by using such WebVR engines (browsers) as Papervision3D, O3D (Google), Silverlight (Microsoft), Virtools, Unity3D, VRP, Torque3D, Quest3D, BS Contact VRML/X3D and so on.

Although WebVR has been applying in various scientific and engineering applications widely and rapidly, the entire WebVR development process of applications is very complicated and difficult for delivering a WebVR application successfully. Some annoying problems are always met during development of WebVR applications as follows:

- Some special restrictions for WebVR applications, i.e. slow transmission and inefficient rendering;
- Uncertainty of WebVR applications development;
- Short time budget to develop a WebVR application;
- Cross-discipline communication in a WebVR project team;
- Lack of appropriate life cycle management methods.

If above problems could not be handled well, WebVR project may be led to frequent delays and accompanied with endless complaints from customers, and even the project fail totally. In order to solve these problems, we introduce agile methodology into the development process of WebVR applications and propose a novel WebVR development model.

The rest of the paper is organized as: Section 2 overviews the related works on agile games development which is similar to agile WebVR development (there are more research results and practical developments on agile games development than agile WebVR development); Section 3 analyzes some major problems and key technologies of WebVR development; Section 4 proposes an agile model for developing WebVR applications; Section 5 explores several cases of developing WebVR projects and shows some performance data; Section 6 concludes this paper and discuss future works of agile WebVR development.

II. RELATED WORKS

To our best knowledge, there is no literature on agile WebVR development found until now. Considering the fact that WebVR development is similar to game development somewhat and there have been some publication on agile game development, we think that some agile game development techniques should be good references for agile WebVR development. Below are the latest works about agile games development:

A few novel pedagogical methods are proposed to improve Location-Based-Mixed-Reality Game (LBMRG) [2], one of them is agile unified process (AUP) to engage students in different thinking styles and continuous evaluation in design. Agile methodology and Scrum framework [3, 4, 5, 6, 7] are introduced into games development [8, 9]. Lean Production [10] is also introduced into the workflow of creating art assets. Test-Driven Development (TDD) [11] is applied to game development in various situations. TDD merits better code design with instant feedback, less documentation and high safety. GM method [12] is presented by combining agile software engineering techniques with inquiry based classroom for teaching game development. Pervasive game prototypes are play-tested using agile development methods of software prototype [13]. Model-Driven Development (MDD) methodology [14] is applied to game development to raise the

Manuscript Received on 21 July， 2010.
E-Maill: jiyuan@gmail.com

level of abstraction for direct and efficient game conceptual modeling. Through studying three known computer games studios, CGS (Singapore), Miko (Singapore) and Goo (London), the agility of game development is explored systematically [15].

III. PROBLEM ANALYSES ON WEBVR DEVELOPMENT

3.1 Problems Encountered During WebVR Development

There are a lot of serious problems which we usually meet during the process of developing WebVR applications. Some of obvious problems can be listed as below:

Web-oriented optimization requirements of VE WebVR applications are embedded into web pages and then viewed by users over the internet. Therefore the developers should do their best to optimize WebVR scenes for real time rendering on Web. These optimizations include mesh reduction, texture reduction, coding optimization and streaming of virtual scenes. WebVR is based on Browser/Server, which is different with traditional networking games based on Client/Server.

Uncertainty The entire process of WebVR development is full of various uncertainties. Most of customers know very little about WebVR initially, they always could not describe their project requirements clearly and completely, there are so many annoying misunderstandings between customers and developers. Only when learning more and more about WebVR, they begin to update their project requirements cutely from time to time. Thus, developers are forced to adjust their technical routes and project planning constantly also, these developers will feel frustrated to redo their works frequently to adapt capricious requirements from customers, and their time and efforts may become wasteful in such an embarrassing situation.

Short time budget Most of customers do not require many complicated functions for their WebVR projects usually, so they think their WebVR project should be finished within a very short period. Moreover, some coming important events or specific dates make customers to shorten their regular project development schedules suddenly.

Difficult cross-discipline communications Developing a WebVR project usually involves programmers, artists, sound synthesizers, animators, project leader and so on. Thus, cross-discipline communications is unavoidable during developing in a project team. Different domain knowledge and background of team members definitely form some communication barriers among them. Therefore, their effective communication is very important to understand each other better and develop WebVR application more smoothly.

Lack of appropriate life cycle management methods As the four problems stated above, WebVR development is quite different with conventional software engineering, also, not same with developing console games. A new life cycle management model is highly expected for more effective supervision of whole WebVR developing process. This is why we decide to study a WebVR oriented life cycle management methods.

3.2 Key Techniques of Developing WebVR Applications

Uncertainty and the *Short Time Budget* are also commonly met in conventional software engineering practice. The other problems are specific for WebVR applications development. In the past 5 years, the following research and development efforts have been doing by our team (*sse.tongji.edu.cn/cgweb*):

Lightweight WebVR Modeling As Internet bandwidth and transmission speed is limited, it's important to minimize the file size of WebVR scenes or Web3D contents. An image-based tree modeling method [16] is proposed to generate lightweight Web3D tree models, which are only several kilobytes in size and very suitable for WebVR applications on Internet.

Lightweight WebVR Interaction Programming Except for WebVR models or Web3D contents on Internet, there are WebVR interaction programs on Internet, which should be also light-weighted and high-efficient. Here, Cohen-Sutherland coding is extended to lightweight programming on interactive operations in WebVR games [17]. An efficient but lightweight navigation algorithm [18] is presented to navigate large scale distributed VRML/X3D environments on Web.

Progressive Downloading Management The mode of WebVR applications is "down-and play". How to get interactive downloading of large scale WebVR scenes on Internet always is a bottleneck problem of DVE. An incremental SMLAOI algorithm is proposed for progressive downloading of large scale VRML/X3D environments progressively [19].

Lightweight Development Model The above three foregoing techniques are critical to break through the first bottleneck problem WebVR applications, "*Web-oriented optimization requirements*". In order to solve other four problems, our team introduces agile methodology WebVR development and makes its development more "agile". The proposed agile "Roll & Raid" development model brought our WebVR team members with lots of benefits through daily practices gradually. In the next section, a brief introduction on agile methodology will be given, then, how "agile" is applied to our developing WebVR applications adaptively and form a novel agile "Roll & Raid" WebVR development model.

IV. AGILE DEVELOPMENT MODEL OF WEBVR APPLICATIONS

4.1 Introduction to Agile Methodology

In 2001, an interest group of iterative and agile methods met together to organize Agile Alliance (www.agilealliance.com) with such "agile manifesto" as *Individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan*, which became a big milestone to agile methodology [20]. The main idea of agile methodology is a collection of methods that tries to be responsive to the needs of the software development process, that is based on practice and experience, and that focuses on being effective and sufficient [21].

Under the umbrella of agile methodology, there are so many concrete agile methods and practices: XP, TDD, FDD, Scrum and so on [22]. A brief introduction of them is summarized as below:

XP (eXtream Programming) is a well-known agile method which is founded on four values: *communication*, *simplicity*, *feedback*, and *courage*. In addition to IID (Iterative and Incremental Development), it recommends 12 core practices: planning game; small, frequent releases; system metaphors; simple design; testing; frequent refactoring; pair programming; team code ownership; continuous integration; sustainable pace; whole team together and coding standards[20].

TDD (Test-Driven Development) is a software development technique that relies on the repetition of a very short development cycle: First the developer writes a failing automated test case that defines a desired improvement or new function, then produces code to pass that test and finally refactors the new code to acceptable standards [23].

FDD (Feature-Driven Design) begins by developing a domain object model in collaboration with domain experts that is then used to create a features list. This is used to produce a rough plan, and informal teams are set up to build small increments over short, say 2-week, periods [22].

Scrum is a management process that can be applied to a number of different activities, not just software development. It contains some groups of practices and predefined roles [22]. The core of Scrum is a set of rules, procedures and practices that are all interrelated, work together to improve the development environment, reduce organizational overheads and ensure that iterative deliverables match the end users' requirements.

4.2 "Roll & Raid": An Agile Process for Developing WebVR Applications

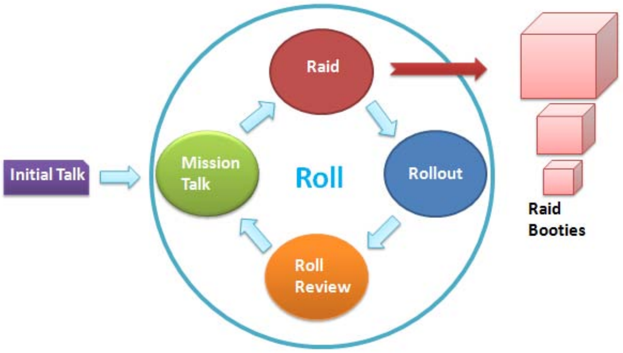


Fig. 1. Overview of "Roll & Raid" process.(Color Plate 9)

In this section, an agile process "Roll & Raid" is proposed for our team to develop WebVR applications. Basically, All the WebVR projects we developed can be classified into 2 genres: WebVR simulations and Web games. Although there are some differences between them when "agile" is applied to, they still share the same idea of "Roll & Raid" process.

The agile process "Roll & Raid" can be described as Fig. 1 and some terms are explained as below:

Initial Talk Initial talk means the first talk when WebVR development team and customers meet together initially when customers desire to develop a new WebVR application. Then, a feeling of familiarity should be established gradually between developers and customers. In addition, it is also very important to convince customers with feasibilities of developing WebVR applications. So developers should show some WebVR demos to customers, and customers can further introduce their specific domain knowledge and project requirements to the developers.

Mission Talk Once a feeling of familiarity between customers and developers was established at "Initial Talk", both sides should have a better understanding of each other's requirements or specialties. Thereby, it is the time for developers and customers to meet again for further "Mission Talk". In this talk, customers should put forward their thoughts and requirements about the to-be-developed WebVR application as many as possible, and then developers also should feedback customers with good advices from the viewpoint of developers. At the end of the phase, developers and customers will reach a consensus on the "missions" which need to be completed at the later phases. Here, "mission" means a set of functions to implement or a group of solutions to find.

Raid With missions obtained from the last phase, developers start to develop the WebVR application. This process is named "Raid" as developers always needs to finish an application with lots of functions within very limited time budget, thus, the agility of the whole developing team is very important as if a spearhead raid enemy's base in a very dangerous and urgent situation. Raid is the core of the "Roll & Raid" process. Its mechanism can be described as the following Figure. 2.

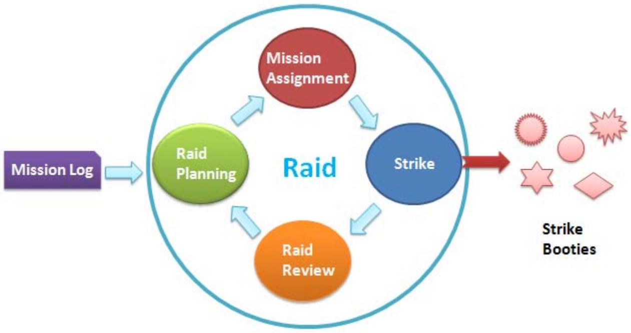


Fig. 2. Raid phase of "Roll & Raid" process

"Raid" starts with a *mission log* which records all missions to develop at the first step. Then, *Raid Planning* is conducted to prioritize for all missions according to their complexity and importance. Then, these prioritized missions will be assigned to different developers in *Mission Assignment* phase. Thus, developer team can start doing these missions and produce lots of *Strike Booties* in *Strike* phase. The strike booties means are the code snippets things, some rough art assets, good ideas or anything else that can be regarded as contributions to these missions. After the strike phase going, developers can stop to review these produced strike booties. If there are still some unaccomplished missions, then goes to next round raid planning; otherwise, ends such raid phase.

Rollout Once all the missions have been finished in the "raid" phase, a application called *Raid Booty* is completed and can be released. The rollout phase is the "show" time when developers demonstrate the newly developed WebVR application. At the same time, developers should also collect those feedbacks from customers, to help developers grasp real attempts of customers and guide them into another developing cycle.

Roll Review In the Roll Review phase, developers should review the newly developed applications, its development process and the feedbacks from customers. If customers accept the current developed applications and they do not want to do anymore, then this roll cycle can be ended. Otherwise, a new mission talk between developers and customers should be initiated with a new roll cycle.

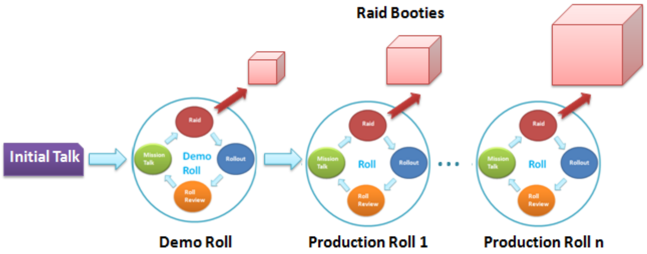


Fig. 3. Actual "Roll & Raid" process

After explaining all of the terms of "Roll & Raid" process above, it is known that the "Roll & raid" involves a series of Roll cycles and Raid cycles. In addition, there is a special Roll cycle named "Demo Roll" in practical WebVR project developments. Demo Roll means the first roll in the entire WebVR "Roll & Raid" process and can be used to create a demo application rapidly to guide communications between developers and customers. Its flowchart can be depicted as Figure. 3. Generally, Demo Roll should be one week's work and every Production Roll may last for 2-4 weeks varying with the complexities of its Mission Log.

4.3 Some Principles of Agile Developing WebVR Applications

"Roll & Raid" is an ideal agile model of developing WebVR applications. In addition, some concrete principles are still required to be abided in daily practical developments. The following principles are summarized by our team for agile development of WebVR applications:

Do not over-consider Mission Log after Mission Talk. After acquiring a Mission Log in Mission Talk, developers should execute the missions in the log as soon as possible. There is really no time budget for developers to over-consider about the missions in the log or even add new missions into the log. If you really want to do so, you need to contact the customers and arrange another mission talk.

Keep it simple. This principle means that developers should write source codes compactly, build lightweight models and design simple user interfaces.

Frequent but short time communications. The first agile manifesto, "Individuals and interactions over processes and tools", means that frequent but short time communications is helpful for team members to understand the whole WebVR applications better and cooperate more smoothly.

Don't make a long-time Roll cycle. Long roll cycle may cause more wastes and less values deployed to customers. Generally, Demo Roll should be one week long and Production Roll 2-4 week long depending on the complexities of the mission log.

V. RETROSPECT AND DISCUSSION

Our team has been developing WebVR applications for about 3 years. In the early stage, three WebVR applications (Fig. 4) were developed by using VRML/X3D. One is Virtual Campus of Tongji University, and the other two projects are Virtual Rocket -Launching System and Smart House System. The first one is a non-profit project which is a start for our team to learn WebVR development. The other two are for digital museum of science and technology to popularize science knowledge.

Initially, we were novices to WebVR development and no appropriate life cycle method is used to guide entire project development process. So we met many troublesome problems and felt painful cutely, which made projects delay for weeks or even months usually. As a project leader, I received endless complaints from customers and team members both. Sometimes, customers ask us to remove some features we implemented with great efforts. It is a horribly huge waste in man power and time for our team. Also, we need to fix lots of bugs when integrating all works from different team members together. All the problems described in Section 3.1 are due to the complexities of WebVR applications development.

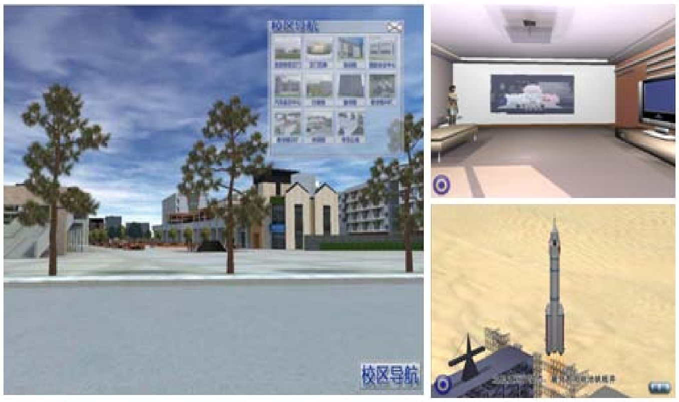


Fig. 4. WebVR applications developed before applying agile methodology

Up until 2009, we started to apply agile methodology into our developing new WebVR applications as shown in Fig. 5. As we expected, all those projects had been finished "agilely" with very good quality. Most important, many suffering and tortures can be avoided during our team developing WebVR projects.

In order to verify the effectiveness of our proposed agile development model, some performance data on 7 WebVR projects are listed in Table 1. For each WebVR application, we list its corresponding *Size*, the *Developer* involved in the development process, the total *Week Cost* to develop that application, the *Man-Week* which is the product of the *Developer* and the *Week Cost*, and a coefficient *MW/S* which represents the quotient of the *Man-Week* and *Size*.

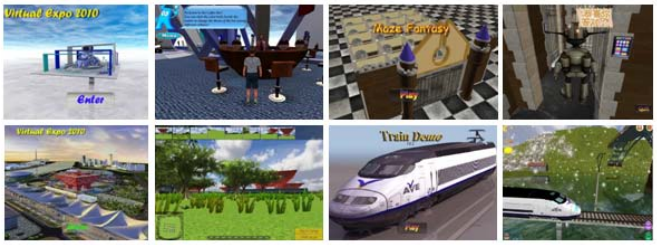


Fig. 5. WebVR applications developed by using our proposed agile development model

To measure *Size* of a WebVR application, we introduced *Key 3D-Model* and *Key Function Point*. A *Key 3D-Model* means a building, a tree, a bridge, an avatar and so on. And a *Key Function Point* means such features as walkthrough, flythrough, interaction, GUI, navigation and so on.

Then, the *Size* can be simply estimated by this way: each *Key 3D-Model* in the scene is scored 2 and each *Key Function Point* of the WebVR application is scored 1, then we can get the *Size* by calculating a sum score of all the *Key 3D-Models* and *Key Function Points*. *Size* is, as a rough estimation, enough to measure the complexity of different WebVR applications.

TABLE 1: WEBVR APPLICATIONS DEVELOPED BY OUR TEAM IN 2009

WebVR Apps	Size	Developer	Week Cost	Man-Week	MW/S
Virtual Campus	54	10	24	240	4.4
Rocket-Launch	20	6	10	60	3
Smart House	30	6	12	72	2.4
Virtual Expo	16	4	4	16	1
Maze Fantasy	9	3	2	6	0.7
Virtual City	12	2	5	10	0.8
Express Train	14	4	4	16	1.1

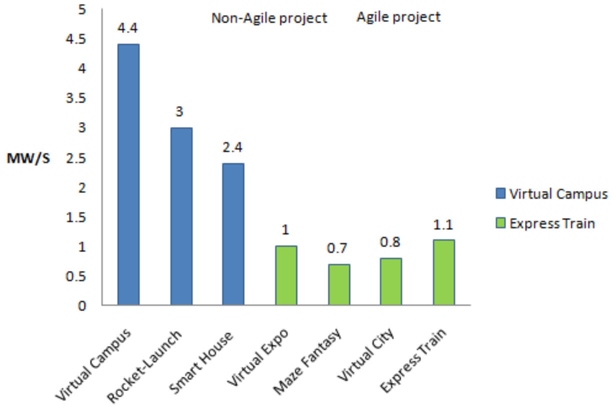


Fig. 6. MW/S of the WebVR applications

Comparing with other parameters, *MW/S* is the best metric to measure the efficiency of developing WebVR projects. And it can be calculated by using (1), where $N_{\text{Key 3D-Model}}$ and $N_{\text{Key Function Point}}$ symbolize the amounts of Key 3D-Model and Key Function Point.

$$MW/S = \frac{Developer \times WeekCost}{2 \times N_{\text{Key 3D-Model}} + 1 \times N_{\text{Key Function Point}}} \qquad (1)$$

The *MW/S* above means how many *Man-Weeks* should be paid to each unit of a WebVR application. The greater it is, the lower the efficiency is. All the *MW/S* values of our 7 WebVR projects are depicted in Fig. 6. The MW/S values of former 3 projects (blue color) without agile model are higher than the ones of latter 4 projects with agile “Roll & Raid” development. The two third of Man-Weeks can be reduced by agile development on the average. So the cost of WebVR projects has been lowered effectively..

VI. CONCLUSION AND FUTURE WORK

In this paper, we present a solution to some problems in developing WebVR applications. Inspired by success of agile game development, we also introduce agile methodology into WebVR development process and propose an agile “Roll & Raid” model of developing WebVR applications. Then, some agile principles are given as a guide to practical WebVR development. The proposed agile “Roll & Raid” model has been applied to develop several WebVR projects already. One year's experience of WebVR development also testified that WebVR can be developed “agilely” in rather shorter period but with good qualities, and our team members can enjoy more fun from such agile WebVR development process.

The proposed agile “Roll & Raid” model is just a good start to apply agile into WebVR applications development. Actually, a lot of other agile principles and practices are required to enhance such agile “Roll & Raid” model. That will become our future work in the next stage. Anyway, this paper has shown that the feasibility of agile WebVR applications development. We believe, with more ideas, more principles and more practices going on in this way, WebVR development will become more and more agile!

ACKNOWLEDGEMENT

This research is supported by Shanghai Key Breakthrough Grant of Science & Technology, Grant No. 08511501001.

We would like to thank those anonymous reviewers for their insightful comments. Also, we sincerely thank our WebVR team members, Ms. Zhao Zhang, Ms. Hong Zhang, Ms. Wenjun Cai, Mr. Chang Luo and Mr. Wenzhao Ren for their kind cooperation and constructive discussion. Furthermore, special thanks should be given to Shanghai Key Breakthrough Grant of Science & Technology (Grant No. 08511501000) for its supports to start this research.

REFERENCES

[1] VRML & X3D, *www.Web3D.org, 2010*.

[2] N.J. Bidwell, and J. Holdsworth, "Battleship by foot: learning by designing a mixed reality game", in *Proceedings of the 3rd Australasian conference on Interactive entertainment*. 2006, Murdoch University: Perth, Australia.

[3] C. Keith, *"Scrum Rising: Agile Development Could Save Your Studio", in Game Developer*. 2007. p. 21-26.

[4] C. Keith, "Scrum and Long Term Project Planning for Video Games". *Gamasutra07* Dec.18, 2007; Available from: http://www.gamasutra.com/view/feature/3142/scrum_and_long_term_project.php.

[5] C. Keith, "Beyond Scrum: Lean and Kanban for Game Developers". *Gamasutra08* Nov.11, 2008; Available from: http://www.gamasutra.com/view/feature/3847/beyond_scrum_lean_and_kanban_for_php.

[6] C. Keith, "Why Use Scrum?", *GameDevNet08*; Available from: http://www.gamedev.net/reference/business/features/whyScrum/page3.asp.

[7] C. Keith, "Advanced Scrum and agile development". *GDC09*, 2009.

[8] C. Keith, "Get in the Game: What others can Learn From Game developers", in *BETTER SOFTWARE*. 2006,

[9] C. Keith, "Agile Methodology in Game Development", in *GDC06*. 2006.

[10] C. Keith, "Lean Game Production". *GDC09*, 2009.

[11] N. Llopis, and S. Houghton. "Backwards Is Forward: Making Better Games with Test-Driven Development", in *GDC06*. 2006.

[12] T. Goulding, Complex game development throughout the college curriculum. *SIGCSE Bull*., 2008. 40(4): p. 68-71.

[13] E.M.I. Ollila, R. Suomela, and J. Holopainen, "Using prototypes in early pervasive game development". In *Computer Entertainments*, 2008. 6(2): pp. 1-17.

[14] E. M. Reyno, and J. A. C. Cubel, "Model-Driven Game Development: 2D Platform Game Prototyping", in *9th International Conference on Intelligent Games and Simulation*, 2008.

[15] P. Stacey, and J. Nandhakumar, "Opening up to agile games development", *Communications of the ACM*, Vol. 51, pp. 143-146, 2008.

[16] Ruoxi Sun, Jinyuan Jia, Hongyu Li and Marc Jaeger, "Image Based Lightweight Tree Modeling for WebVR Application", *ACM SIGGRAPH VRCAI09*, 2009, Japan.

[17] Zheng Li, Yi Sun, Jinyuan Jia, "Extension of Cohen-Sutherland Coding and Its Application to Light-Weighted VRML Programming", *IADIS International Conference Web Virtual Reality and 3DWorld, Algarve, Portugal*, pp. 19-21 June, 2009.

[18] Jinyuan Jia, Guanghua Lu and Yuan Pan, “An Efficient Navigation Algorithm of Large Scale Distributed VRML/X3D Environments”, *Lecture Notes in Computer Science, LNCS*, Vol. 3434, pp. 244-252, Springer-Verlag August 2007. HCII07, Beijing, August, 2007.

[19] Jinyuan Jia, Ping Wang and Sen Wang, Yuying Wang, “An Integer Incremental AOI Algorithm for Progressive Downloading Large Scale VRML/X3D Environments”, *Lecture Notes in Computer Science, LNCS*, Vol. 3522, pp. 711-722, Edutainment07, Hong Kong, June, 2007.

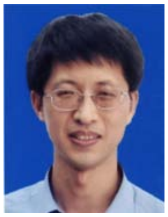
[20] Craig Larman. "Agile and Iterative Development", *Addison Wesley*. 2003.

[21] John Hunt. "Agile software construction", *Springer-Verlag London Limited*. 2006.

[22] Mike Holcombe. "Running an Agile Software Development Project", *WILEY*. 2008.

[23] Beck, K. "Test-Driven Development by Example", *Addison Wesley*. 2003.

Zhixing Shen is a postgraduate student with the School of Software Engineering, Tongji University, Shanghai, China. His research interests include agile development on games and WebVR applications, Character Animation and Game Engine.



Jinyuan Jia is a professor with the School of Software Engineering, Tongji University, Shanghai, China. He is chairing a research center of graphics and images. His research interests include WebVR, Game Engine, Virtual Reality, Geometry Modeling and Computing, CAD and so on.