STUDY ON MANAGEMENT ALTERNATIVES ON AGILE METHOD BASED FINANCIAL SECURITY SOFTWARE PROJECT

[1]YONG-GUK JEONG, [2*]KOO-RACK PARK, [3]DONG-HYUN KIM

[1]Dept. of Multimedia Engineering, Kongju National University, 31080, Rep. of Korea

[2*]Dept. of Computer Science & Engineering, Kongju National University, 31080, Rep. of Korea

[3]Dept. of Computer Engineering, Kongju National University, 31080, Rep. of Korea

E-mail: [1]lucifur1@gmail.com, [2*]ecgrpark@kongju.ac.kr, [3]dhkim977@naver.com

ABSTRACT

In tandem with development of information technologies, telecommunication environment, infrastructure and device etc. are getting smarter and converged. As non-facing service transactions are expanded, financial institutions provide internet banking service, which customers are using with convenience. Accordingly, the information system market related with finance is anticipated to experience much changes in system setup due to change in business and financial environment, and upgrade or new development is continuously performed. However due to the development of informatized society, hacking and virus attack technics are becoming sophisticated and diversified, and also as the cases of infringement of personal privacy etc. are continuously occurring, much efforts are required to resolve these situations. Especially, the current attack paradigm on electronic financial service is concentrated on the customers using electronic finance, so the need for security system is ever increasing to resolve this situation. However, in pursuit of any effort to resolve such situation by proceeding financial security related project, mostly the time and cost incurred for the project is burdensome. Especially, most of such projects accompany numerous risks, and representatively any change in requirements of customers is exercising so heavy influence as to determine the success of the project. Any change in requirements related to financial security shall be immediately processed, but delay of project often occurs due to poor communication between developer and customer. In this regard, this thesis proposes the system to compose development team based on Agile Methods, to differentiate the role of team members and manage the project period, in case financial security software project is performed while development period and development cost is fixed. Also in order to verify the proposed system, actual project is implemented and comparison is made with the waterfall model which is the traditional development methodology. The result of verification shows effect of reduced number of 8% days required for development saving effect. By the proposed system, it is expected that successful progress of financial security related software project would be possible. In the future, subsequent studies on flexible expandability and systematic management methodology are necessary.

Keywords: *Cloud Computing, Project Management, Agile Methods, Scrum, ICT*

1. INTRODUCTION

Recent development of information technologies is exercising great influences on the development of telecommunication environment, infrastructure, device, and are on the gradual trend to convergence and are becoming smarter [1].

Also financial institutions are evolving from customer facing oriented service to non-facing transactions, and are providing various services such as internet banking, smartphone banking etc. Also service delivery channels by electronic means such as internet etc. are providing convenience and flexibility to customers of banking services [2].

As such, the finance related market in modern society is expected to experience various changes in setting up electronic financial information system due to change in business and environment. And upgrade or new development of already existing information system is the trend with continued progress. Each financial institution is continuously making electronic finance related investment, and thanks to the development of internet and smart devices, realization of informatized society is being realized in acceleration.

However due to development of informatized society, attack technics such as hacking and virus etc. are getting more sophisticated and diversified such as the study to prevent hacking through bluetooth and smartphone virus in case of email transmission with various files attached or including internet link which automatically installs hacking program upon click, and much efforts are devoted to resolve these situations[3],[4],[5].

Especially, the current attack paradigm on electronic financial service is concentrated on the customers using electronic finance, provoking viruses, hacking and the elements jeopardizing personal privacy and electronic financial transactions, so the need for security system for electric finance to resolve this situation is ever increasing [6].

Recently, ransomware security issues such as malware has appeared [7],[8], and security algorithm also varies moment by moment [9],[10]. To resolve this, the financial industry tries to proceed with financial security related projects, but the time and cost incurred for the project is mostly burdensome.

Accordingly, application of traditional development methodology whereby everything is planned for design and implementation at initial stage of project, may provoke the occasion where project progress becomes very difficult. To resolve such projects, various development methodologies have been proposed.

In this regard, this thesis proposes the system to compose development team based on Agile method, to differentiate the role of team members and the management system to check the project period, in case financial security software project is performed while development period and development cost is fixed. By the proposed system, it is expected that successful progress of financial security related software project would be possible.

2. RELATED RESEARCH

2.1 Agile methodology

Agile development methodology is a much lightweighted methodology in comparison with the waterfall development methodology which is the traditional software development methodology, and the core is in satisfying the customer who gives order for software and deriving the success of business. This mean running software rather than document, personal interaction rather than tool, adaptation to change rather than proceeding as planned, and the method to agree to Agile principle such as customer cooperation rather than negotiated contract [11].

Core concept of Agile development method can be defined as human focused and adaptation [12], and based on this concept new development methods such as scrum and extreme programming etc. have been presented. These days, scrum is most often adopted and used. With merits of enhanced quality and productivity, accommodativeness to change, quick market entry time, easy management of adjustment in priority, use of scrum is on abruptly growing trend in the software development industry [13].

However agile methodology regards the investment for future reuse in the abruptly changing environment as a risk element, and hence is half-hearted in development of reusable outputs [14].

The scrum methodology is mainly used as management and development methodology in the agile software development, as the framework used for development of products and project with gradual and repeated characteristics [15].

Especially this is the resolution method for small scale team focused project whereat change in requirement occurs frequently [16]. Generally scrum process comprises project preparation and plan as project plan stage, release plan, sprint plan, daily work, sprint review and retrospection, release review as development stage, and then project closing process as the project finishing stage

2.2 The Waterfall SDLC Model

The following [Figure 1] is the process of waterfall development method [17].

The waterfall development method is the sequential software development process, composed mainly of 5 stages.

First is the analysis stage, where explanation of the movement of the software to be developed (to be defined by system and business analysts), functional, non-functional requirements, interaction between software and user are explained.

Second is the design stage, where software solution is planned and problems are resolved with algorithm, software architecture, database design, Graphic User Interface design, and data structure definition included in the design.

Third is the development stage, where definite program is made realizing business requirements and design specification, meaning the realization of composition elements of database, software and website etc. This is the stage to convert the overall requirement into production environment.

Fourth is the test stage, which is called as 'effectiveness inspection' whereby it is verified whether the requirements and specifications have been satisfied and the intended purpose has been achieved. This test determines whether product at the development stage satisfies the requirement stipulated at the commencement of the relevant stage. Also verification is the process to evaluate the software, in order to determine whether the software would satisfy the particular requirement when the development process is completed [18].

The fifth is the maintenance stage, where software solution is corrected at or after distribution, for input, error correction, enhancement of functionality and quality. Also this stage is used to accommodate new user requirements and enhance the confidence on software [19].

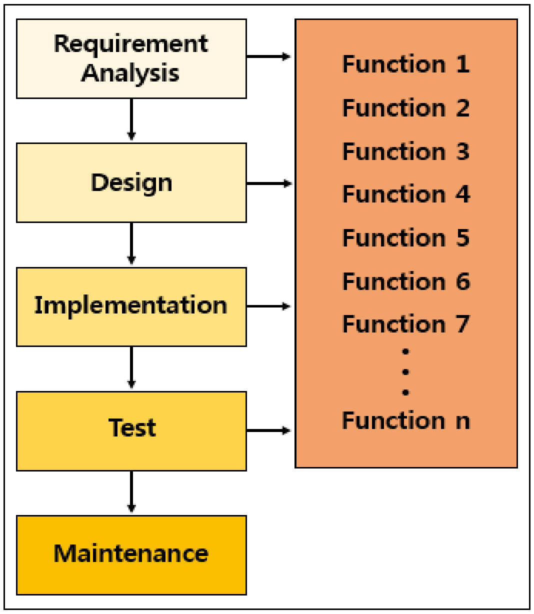


Figure 1: Waterfall Development Process

2.3 Scrum

Scrum was developed to manage the process of system development project. The idea of industrial process control theory is applied to system development projects to access empirically approaches to reintroduce the idea of flexibility, productivity, adaptability [21]. However, Scrum is not a specific software development technology for the implementation of software in development projects. Scrum is to focus on what the members of the development team work to flexibly produce continuously changing systems.

The main point of Scrum is related to environmental variables with some technical variables (time, resources and technology, requirements) that can be changed in the course of system development. Therefore, it is impossible to predict the development process, it becomes complicated, the flexibility of the development process of the system that can cope with customer's change is necessary, and a system convenient when done as a result of the development process is produced [22].

Scrum has given a lot of help to improve the existing engineering practices within the organization. Also, since scrum is aimed at continuously identifying obstacles and defects in the development process and methods applied to the project, frequent management activities are necessary [23].

The following [Figure 2] is a scrum development methodology.

The difference between the existing waterfall model, spiral model and scrum is as follows. Scrum assumes that the sprint stage analysis, design and development process can't predict in advance. Also, the control mechanism is used to manage unpredictable parts and control risks. Characteristics of Scrum can be roughly divided into four as follows [24].

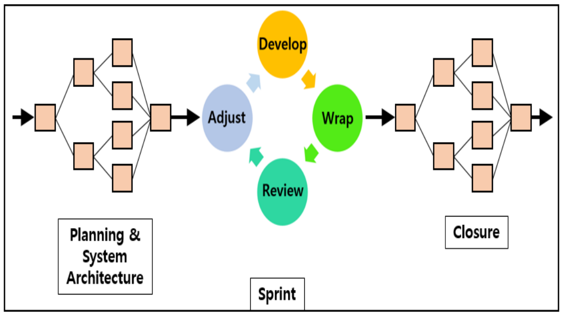


Figure 2 : Scrum Methodology

First, in the start and end phases (planning and closing phases), the processes are configured with I / O definitions for all processes, and at the planning stage, there are several iterations.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Second, the sprint step is a process of experience, many processes not specified or controlled. In other words, it is handled as a black box that requires external control, so maximizing flexibility and control including risk management are repeatedly applied at each sprint stage.

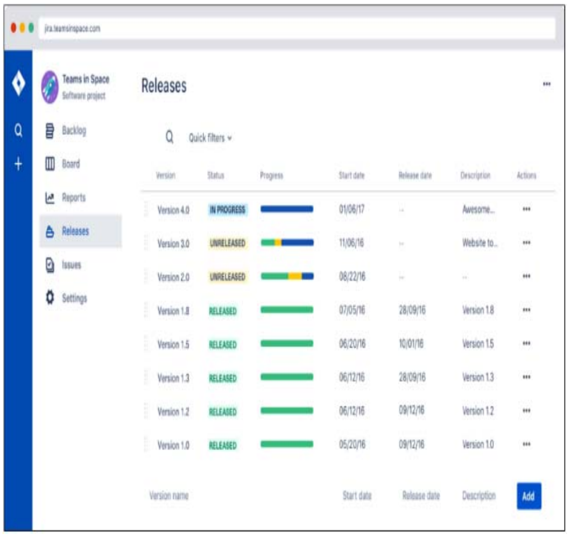
Third, sprint has a nonlinear structure and is very flexible. Where possible, explicit process knowledge is used, if impossible implicit knowledge and trial and error are used to build knowledge. Therefore sprint will eventually be used to develop and evolve the product.

Fourth, it is open to the environment until the completion stage of the project. The project deliverables are easy to change at the project planning and sprint stage, flexible in terms of environmental complexity such as time and quality, pressure on development costs. Also, the deliverables are determined according to the environment while the project is in progress.

2.4 JIRA Software

Jira Software is the issue tracking system developed by Atlassian Company, and is the tool used by agile team to support project management using agile methodology [20].

The following [Figure 3] is the On-Premise of jira software.



jira.teamsinspace.com

Teams in Space
Software project

Backlog
Board
Reports
Releases
Issues
Settings

Releases

Quick filters

Version	Status	Progress	Start date	Release date	Description	Actions
Version 4.0	IN PROGRESS		01/06/17	--	Awesome...	•••
Version 3.0	UNRELEASED		11/06/16	--	Website to...	•••
Version 2.0	UNRELEASED		08/22/16	--	--	•••
Version 1.8	RELEASED		07/05/16	28/09/16	Version 1.8	•••
Version 1.5	RELEASED		06/20/16	10/01/16	Version 1.5	•••
Version 1.3	RELEASED		06/12/16	28/09/16	Version 1.3	•••
Version 1.2	RELEASED		06/12/16	09/12/16	Version 1.2	•••
Version 1.0	RELEASED		05/20/16	09/12/16	Version 1.0	•••

Version name Start date Release date Description Add

Figure 3: On-Premise

The function provided presents the backlog, board, reports, releases, issues, settings. Backlog is the collection of issues not included in the releases. Board comprises kanban board and scrum board. The flexible kanban board ensures full appreciation of the works to be performed subsequently, and enables derivation of maximum results continuously within the minimum cycle time.

The scrum board makes description to focus on creating the maximum values that repeat promptly and increase. Release is the stage where the issues using repeated agile method are allocated. Each object allocated from backlog to releases is named as an issue, which has the function for making registration and correction.

3. PROPOSED SYSTEM

3.1. Use of Agile methodology

The following [Figure 4] is the composition chart applying the Agile methodology to the financial security software project proposed in this thesis.

The proposed methodology is the method to complete the project within designated period in case the development period and cost have been fixed in the financial security software development project, rather than applying repeated development methodology only to agile scrum.

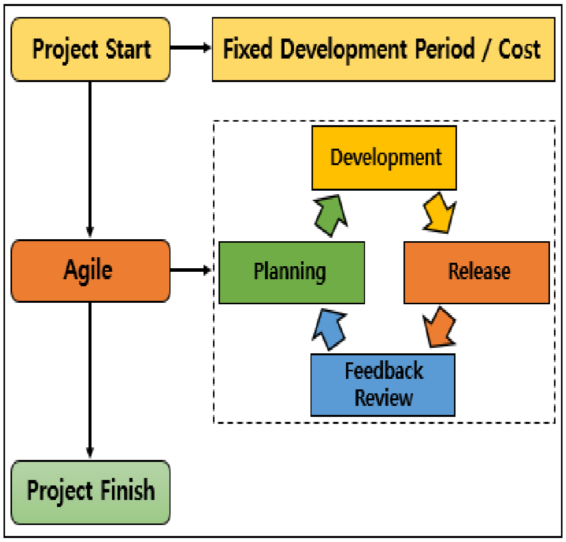


Figure 4 : Utilization of Agile Methodology

3.2. Composition and role of team

In order to complete the project within project period while using scrum process, composition of scrum team is a very important element.

The scrum team is composed of developer and customer, designer, senior developer. The following [Table 1] shows the composition and role of the scrum team.

Table 1: Organization of Team and Their Roles Application Architecture

Composition	Role
Developer	- Developer with superior capacity - Cooperation is emphasized
Designer	- UX/UI designer - Immediate response to requirements of user side and customer
Senior developer	- Developer and tester - Technical review on the direction for change - Guide provide in consideration of refactoring
Customer	- Perception that cost/period is fixed - Requirements presented according to trend

Cooperation based scrum team shall observe the following 5 principles.

Firstly, scrum team shall freely speak and discuss the idea derived through repeated releases without any restriction in role.

Secondly, developer and designer shall understand any change in the realized program in response to ever changing environment.

Thirdly, senior developer shall derive improvements on reference materials suitable for ever changing environment as well as on existing developments.

The fourth, customers shall sufficiently perceive that the cost/period are fixed, and promptly determine whether it is possible or not.

The fifth, continued improving measures shall be pursued, without any concern on failure.

3.3. System composition

3.3.1. System architecture

The following [Figure 5] is the architecture of the proposed system.

Interface server is composed for connection between Jira server and the management server of the proposed system. Through this, the initial Agile related data at Jira server is collected and processed and then transmitted to management server.

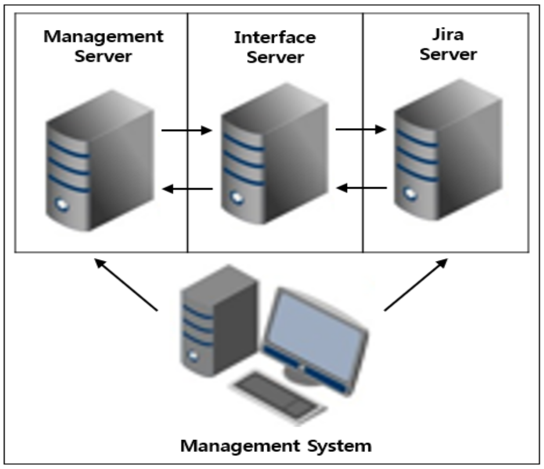


Figure 5 : System Architecture

3.3.2. Application Architecture

The following [Figure 6] is the application architecture.

Web server composes 1 tiar using apache 2.2, and 2 tiar WAS (Web Application Server) used tomcat8.5 version. And DB is composed at 3tier, using mariaDB 5.5.

Framework is the light weighted one, using spring4 that can use java and restfull method, using jersey 2.26 for composition of interface. Also in order to process Jira data, spring batch job is composed.

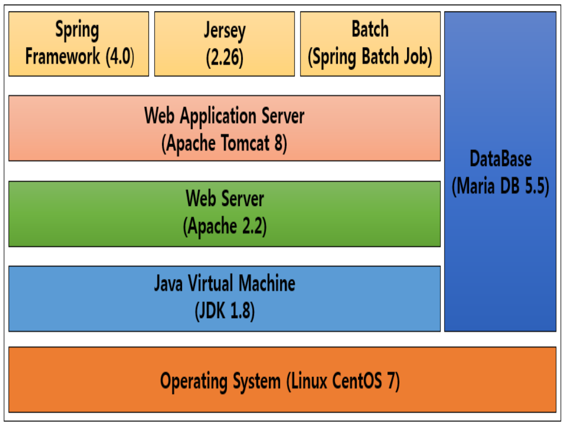


Figure 6 : Application Architecture

3.4. Configuration of menu

The following [Figure 7] shows the configuration of menu for the proposed system. As shown in the following figure, the menu structure can be roughly divided into three as Dashboard, WBS (Task Integration Management), LINK.

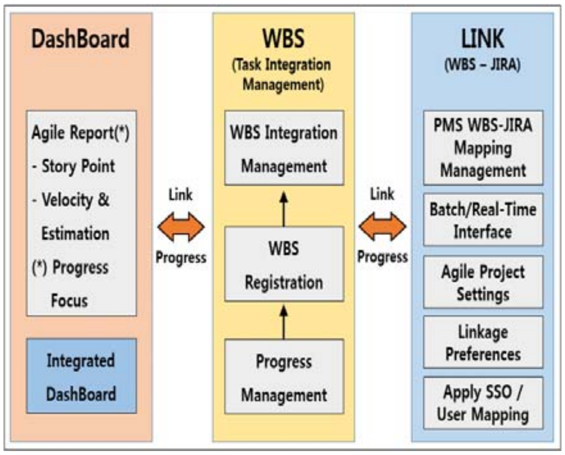
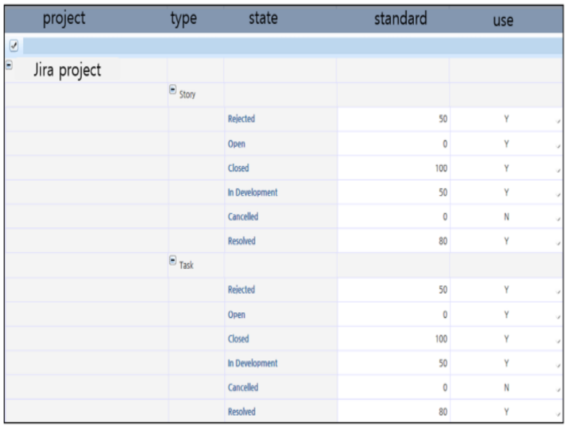


Figure 7 : Menu Configuration

3.4.1. LINK

The following [Figure 8] shows the types of link.



project	type	state	standard	use
☑				
Jira project				
	Story			
		Rejected	50	Y
		Open	0	Y
		Closed	100	Y
		In Development	50	Y
		Cancelled	0	N
		Resolved	80	Y
	Task			
		Rejected	50	Y
		Open	0	Y
		Closed	100	Y
		In Development	50	Y
		Cancelled	0	N
		Resolved	80	Y

Figure 8 : Progress standard

For the link, mapping information is composed by interface between Jira server and composition server of the proposed system, and data is processed by real time Interface with batch. Link is composed in classification of 5 categories like the setting on agile project, setting of link environment, SSO (Single Sign-On) application.

First is the Mapping management between WBS of the Project Management System (Work Breakdown Structure) and Jira Server, which manages mapping of project key data selected as the standard at the link system between the proposed system and Jira server, with Sprint as the data used at Jira.

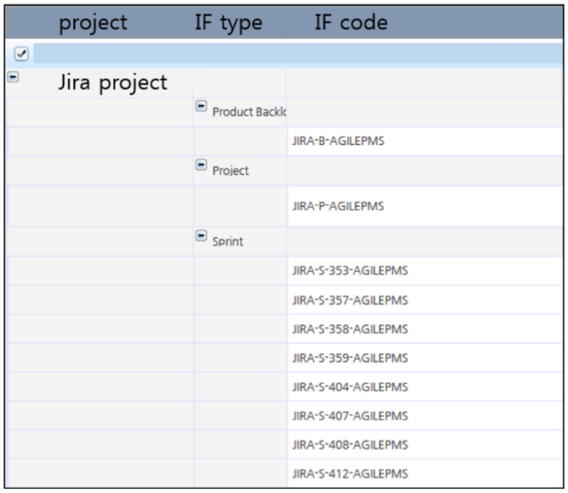
Second is the Batch/Real-Time Interface, which is the procedure to prevent server load and enhance the accuracy of data. As the data quantity varies according to each Sprint, a Batch replaces overall data at regular time, and real time Interface is composed for synchronization with current data before the cycle whereat the batch is implemented. Also batch holds the computation function on the accumulated progress ratio, and regularly updates the user information.

The third is the Agile Project setting, where the project name, project manager in charge, project commencement date, completion date, project user information to be used at the proposed system can be registered.

Fourth, the link composition performs registration on information of the server to be linked, and can compose information of the server to be linked by registering address, access port and peculiar byname of Jira server.

The fifth is composition of SSO (Single sign-on), which implements mapping of information between JIRA user and the proposed user. By applying SSO to two different web sites, mapped user information is composed so that two web sites may be used through single login. This is the function which is capable of mutual mapping for user information necessary for composition of SSO.

The following [Figure 9] is the interface list between Jira and the proposed system.



project	IF type	IF code
☑		
Jira project		
	Product Backlc	
		JIRA-B-AGILEPMS
	Project	
		JIRA-P-AGILEPMS
	Sprint	
		JIRA-S-353-AGILEPMS
		JIRA-S-357-AGILEPMS
		JIRA-S-358-AGILEPMS
		JIRA-S-359-AGILEPMS
		JIRA-S-404-AGILEPMS
		JIRA-S-407-AGILEPMS
		JIRA-S-408-AGILEPMS
		JIRA-S-412-AGILEPMS

Figure 9 : Interface List

Among the types of issues in interface with Jira, progress situations exist in multiple due to high freedom. As progress can be computed only by setting standard for each progress stage, set shall be performed at the proposed system. This composition is implemented so that setup of progress stage by each stage and selection on whether relevant stage should be used, can be made. Even though setting of initial data as it is at Jira is possible, composition is performed so that alteration according to characteristics of the manager and project would be possible.

This is the screen showing the relevant composition after composition of link between Jira project and the proposed system. At the relevant list, the only code value is created that was composed upon link to the proposed system. When composing the relevant code value, a certain rule is engaged.

The relevant rule would use the key repeatedly created upon initial link at the linked system name + index + the proposed system. This is used for the purpose of preventing duplication and maintaining sole code system by creating peculiar code per project. Also keeping sole code system helps in keeping basic security.

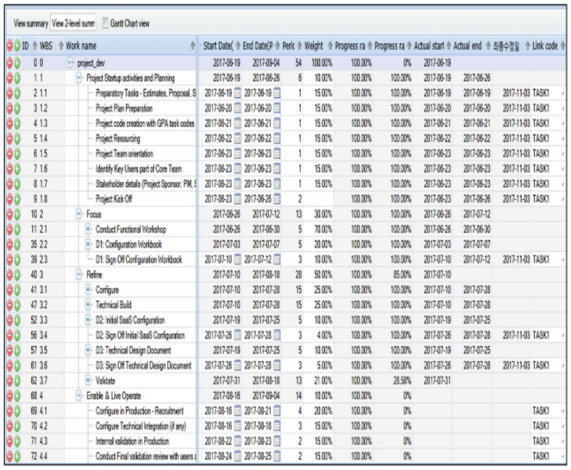
3.4.2. Dashboard

This is the screen showing the summarized project progress, which is advanced based on the data link between the proposed system and the Jira server.

Here, the progress status of agile project and the accumulated status of story points can be confirmed daily and monthly, separate from the biweekly sprint period used at Jira server. Even though Jira uses biweekly repeated system, the proposed system makes divisions into daily figures and describes by monthly sum, detailed progress status can be grasped.

3.4.3. WBS (Work Breakdown Structure)

The following [Figure 10] is the WBS of the proposed system. WBS integrates and shows the major work plan and Jira task link, and manages progress ratio.



ID	WBS	Work name	Start Date(	End Date(P	Peric	Weight	Progress ra	Progress ra	Actual start	Actual end	최종수정일	Link code
0	0	project_dev	2017-06-19	2017-09-04	54	100.00%	100.00%	0%	2017-06-19			
1	1	Project Startup activities and Planning	2017-06-19	2017-06-26	6	10.00%	100.00%	100.00%	2017-06-19	2017-06-26		
2	1.1	Preparatory Tasks - Estimates, Proposal, S	2017-06-19	2017-06-19	1	15.00%	100.00%	100.00%	2017-06-19	2017-06-19	2017-11-03	TASK1
3	1.2	Project Plan Preparation	2017-06-20	2017-06-20	1	15.00%	100.00%	100.00%	2017-06-20	2017-06-20	2017-11-03	TASK1
4	1.3	Project code creation with GPA task codes	2017-06-21	2017-06-21	1	15.00%	100.00%	100.00%	2017-06-21	2017-06-21	2017-11-03	TASK1
5	1.4	Project Resourcing	2017-06-22	2017-06-22	1	15.00%	100.00%	100.00%	2017-06-22	2017-06-22	2017-11-03	TASK1
6	1.5	Project Team orientation	2017-06-23	2017-06-23	1	15.00%	100.00%	100.00%	2017-06-23	2017-06-23	2017-11-03	TASK1
7	1.6	Identify Key Users part of Core Team	2017-06-23	2017-06-23	1	15.00%	100.00%	100.00%	2017-06-23	2017-06-23	2017-11-03	TASK1
8	1.7	Stakeholder details (Project Sponsor, PM, S	2017-06-23	2017-06-23	1	15.00%	100.00%	100.00%	2017-06-23	2017-06-23	2017-11-03	TASK1
9	1.8	Project Kick Off	2017-06-23	2017-06-26	2		100.00%	100.00%	2017-06-23	2017-06-26	2017-11-03	TASK1
10	2	Focus	2017-06-26	2017-07-12	13	30.00%	100.00%	100.00%	2017-06-26	2017-07-12		
11	2.1	Conduct Functional Workshop	2017-06-26	2017-06-30	5	70.00%	100.00%	100.00%	2017-06-26	2017-06-30		
35	2.2	D1: Configuration Workbook	2017-07-03	2017-07-07	5	20.00%	100.00%	100.00%	2017-07-03	2017-07-07		
39	2.3	D1: Sign Off Configuration Workbook	2017-07-10	2017-07-12	3	10.00%	100.00%	100.00%	2017-07-10	2017-07-12	2017-11-03	TASK1
40	3	Refine	2017-07-10	2017-08-18	28	50.00%	100.00%	85.00%	2017-07-10			
41	3.1	Configure	2017-07-10	2017-07-28	15	25.00%	100.00%	100.00%	2017-07-10	2017-07-28		
47	3.2	Technical Build	2017-07-10	2017-07-28	15	25.00%	100.00%	100.00%	2017-07-10	2017-07-28		
52	3.3	D2: Initial SaaS Configuration	2017-07-19	2017-07-25	5	10.00%	100.00%	100.00%	2017-07-19	2017-07-25		
56	3.4	D2: Sign Off Initial SaaS Configuration	2017-07-26	2017-07-28	3	4.00%	100.00%	100.00%	2017-07-26	2017-07-28	2017-11-03	TASK1
57	3.5	D3: Technical Design Document	2017-07-19	2017-07-25	5	10.00%	100.00%	100.00%	2017-07-19	2017-07-25		
61	3.6	D3: Sign Off Technical Design Document	2017-07-26	2017-07-28	3	5.00%	100.00%	100.00%	2017-07-26	2017-07-28	2017-11-03	TASK1
62	3.7	Validate	2017-07-31	2017-08-18	13	21.00%	100.00%	28.58%	2017-07-31			
68	4	Enable & Live Operate	2017-08-16	2017-09-04	14	10.00%	100.00%	0%				
69	4.1	Configure in Production - Recruitment	2017-08-16	2017-08-21	4	20.00%	100.00%	0%				TASK1
70	4.2	Configure Technical Integration (if any)	2017-08-16	2017-08-18	3	15.00%	100.00%	0%				TASK1
71	4.3	Internal validation in Production	2017-08-22	2017-08-23	2	15.00%	100.00%	0%				TASK1
72	4.4	Conduct Final validation review with users a	2017-08-24	2017-08-25	2	15.00%	100.00%	0%				TASK1

Figure 10 : Work Breakdown Structure

Three functions are available, comprising WBS progress ratio management, WBS registration, WBS integration management. The progress ratio management can compute the linked data real time applying the progress ratio computation formula to the proposed system, and WBS registration is possible directly on Excel upload and screen. Also WBS integration management integrates the linked Jira data to enable confirmation of sprint and WBS task.

3.5. Schedule management

The schedule management integrates the jira data linked to schedule plan, and describes the plan progress ratio and performance progress ratio.

The following [Figure 11] shows the progress ratio of performance vis-a-vis plan.

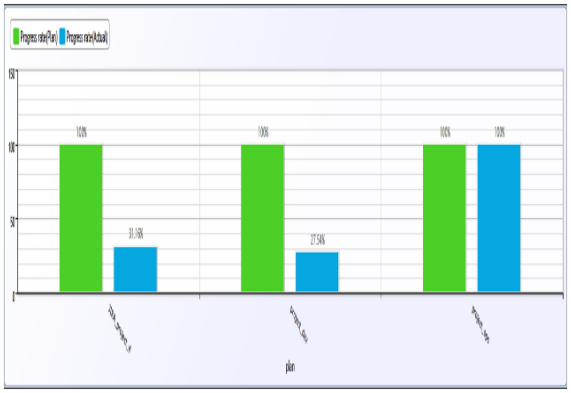


Figure 11 : plan/performance progress ratio

The following [Figure 12] shows the progress ratio of daily progress vis-a-vis plan.

The daily description of plan progress ratio and performance progress ratio, and facilitates easy grasp of performance vis-a-vis plan.

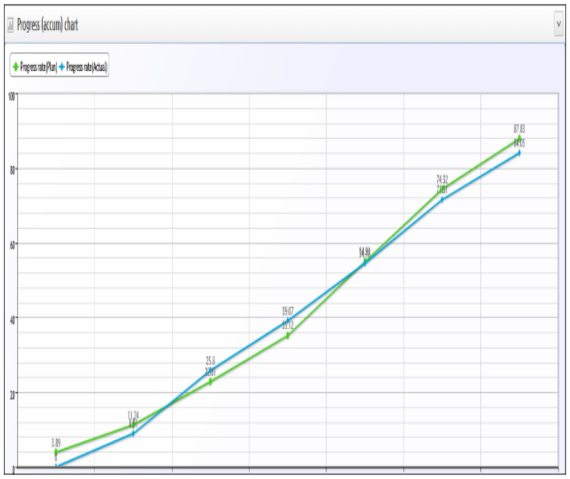
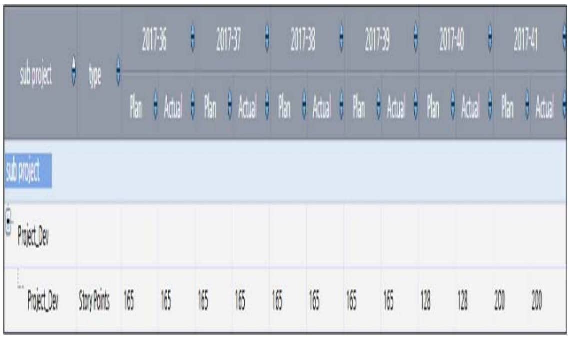


Figure 12 : Daily plan/performance progress ratio

The following [Figure 13] is the graph describing the accumulated plan progress ratio and performance progress ratio.



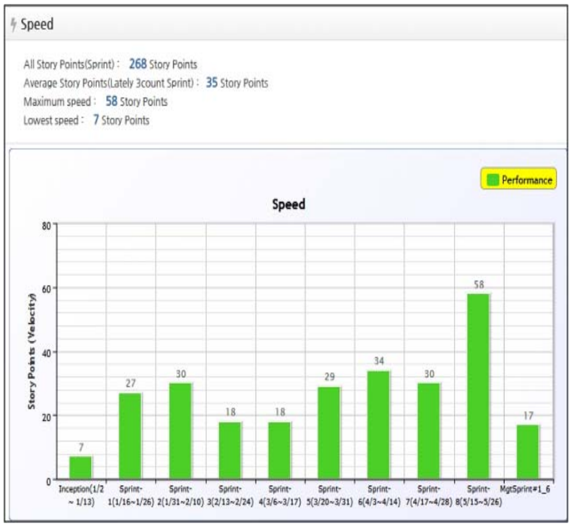
sub project	type	2017-36		2017-37		2017-38		2017-39		2017-40		2017-41	
		Plan	Actual	Plan	Actual	Plan	Actual	Plan	Actual	Plan	Actual	Plan	Actual
sub project													
Project_Dev													
Project_Dev	Story Points	165	165	165	165	165	165	165	165	128	128	200	200

Figure 13 : Weekly plan/performance accumulated

The following [Figure 14] is the screen to capture the velocity of Story point.

Story point is computed as the number of work days, in consideration of the size, complexity and risk level of relevant task. By description of the accumulated story points, the trend of performance can be captured. Sprint is set at units of fortnight, and goes through the procedure of plan, development, release and review.

When this Sprint is repeatedly proceeded with, the score of the sum of planned Story points and the actually achieved Story points are described weekly. This facilitates easy analytic review on any cause of problem, by checking whether development team is proceeding with the Sprint smoothly and by raising any problem.



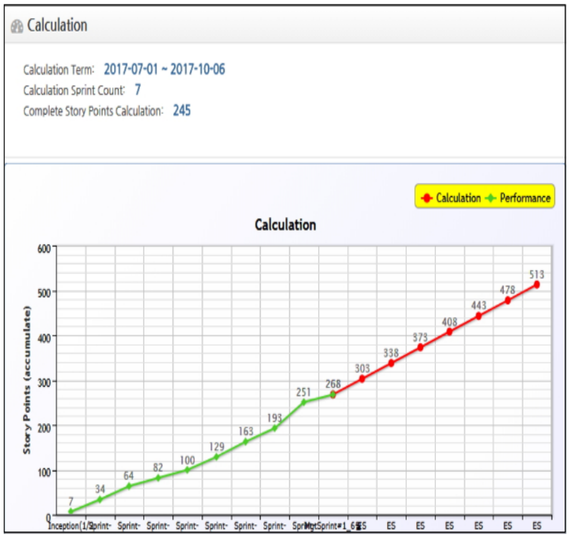
Speed

All Story Points(Sprint) : 268 Story Points
Average Story Points(Lately 3count Sprint) : 35 Story Points
Maximum speed : 58 Story Points
Lowest speed : 7 Story Points

Figure 14. Story Point Speed

The following [Figure 15] is the screen confirming Sprint, which verifies the sprint that achieved the highest story point vis-a-vis the period using total accumulated story point.

As periodic result is provided facilitating easy comparison between relevant periods, useful access to progress management is available. Thanks to the accumulated story points, velocity can be measured and the performance can be estimated based thereon.



Calculation

Calculation Term: 2017-07-01 ~ 2017-10-06
Calculation Sprint Count: 7
Complete Story Points Calculation: 245

Figure 15. Sprint confirmation screen

The progress ratio can be derived from the portion at the workflow eligible as Jira. Workflow represents the progress status on issues. From 1st stage to 4th stage or customarily, progress stage can be described. This portion is the part requiring careful handling when linking.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Relevant stages are involved in multiple numbers, but ultimately can be categorized in 4 classes of To Do, In Progress, Resolved, Done. By addition of the function setting weight in this part, the progress ratio can be derived as the weight according to the relevant stage as follows (formula 1).

$$N_{task} \times S_p \times W = 100 \qquad (1)$$

Here, N_{task} is the number of tasks. S_p is the story point, W is weight, with the total of all of these at 100. With the weight derived by these interface, the performance vis-à-vis plan can be described

At jira the current stage and completed status only can be grasped. In the course of the progress of overall plan, it is not easy for the manager to check the progress stage generally.

4. EXPERIMENTS AND DISCUSSION

In order to verify the system integrating the agile proposed in this thesis and the traditional management methodology, application is made to a project related to financial security project actually proceeded.

The following [Table 2] is the comparison of number of days required at completion of project.

According to the result of comparison between the number of days required for progress of project in case of applying agile management methodology and applying the proposed system, with the case of applying the traditional development methodology set as 100%, the proposed system could bring reduction effect of number of days required by average 8% than in case of applying agile development methodology.

Table 2: Comparison of the number of days required for project

Item	Agile	Proposed System	Reduction Effect (%)
Legacy DRM System upgrade	96	89	7
Enterprise Management System	98	95	3
Card Enterprise Security IF Management	90	84	6
Finance Enterprise Security Management	89	75	14
Insurance Access Control system	91	78	13
Insurance privacy system private cloud switch	78	73	5

From this, the expedition effect of period may be obtained by deriving problems by stage earlier.

Given the effects of shorter periods, it can be seen that the parts of the proposed system that have been utilized are different from the systems that have only the agile methodology.

The effect of shortening the period will have a great impact on the team members in the proposed system. The difference between the proposed system and the agile methodology in the composition of the team is for senior developers.

It can be seen as a result of senior developers participating in the project. It would be different if the project was carried out in a different team structure than if the project was carried out in the first and second divisions using the agile methodology and the proposed system.

If the team members are familiar with this practice, the shorter the time period is more effective. The Enterprise Management System, and Finance Enterprise Security Management projects in [Table 2] are cases in which different projects are carried out with the same teaming configuration.

Legacy DRM system upgrade, Card Enterprise Security IF Management project is a project that has been conducted in a different team.

Insurance access control system project is a case in which a senior developer who has used the proposed system participates in another project.

5. CONCLUSION AND FUTURE DIRECTIONS

In tandem with development of IT industry, the overall society is experiencing various changes by being smarter and converged. Financial institutions are also migrating from customer facing oriented business to non-facing transaction services such as internet banking, smartphone banking.

These service delivery channels by electronic means provide customers with convenience and flexibility, and hence is anticipated to expand abruptly. However, electronic financial service through network such as internet provokes the elements that jeopardize electronic finance-related transactions such as virus, hacking, person privacy.

Each financial institution is accelerating the development and distribution of security system to resolve this issue. However, in the pursuit of financial security related projects to resolve such issues, the time and cost incurred for the project is mostly burdensome. Especially most of the financial security software development projects face limitations in development period and development cost budget.

Also most projects accompany numerous risks, so it is no exaggeration to say that the case of completion within the designated development period is almost non-existing. And representatively, any change in requirements of financial institutions is exercising so heavy influence as to be determinant on the success of the project.

Especially any change in requirements related to financial security by financial institutions shall be immediately processed, but delay of project often occurs due to poor communication between the developer and customer. In this regard, this thesis proposes the management system to control the project development period applying the agile method, to reduce the delay in project development period.

Also the proposed composition of team to resolve this is implemented so that each team member could be faithful to each one's role. Adding senior developer to the team member would be advantageous to the determination on additional application of abruptly changing security algorithm.

Through this, the communication among diverse stakeholders may become smooth, and the basic agile theorem enhancing customer satisfaction may be pursued. Especially scrum development method emphasizes cooperation with customer rather than contracts, and would satisfy the customers by repeated release of requirements that have high potentiality of continuous change, and it is expected that successful progress of financial security related software project would be possible.

As the direction of future studies, subsequent studies are necessary on flexible expandability and systematic management methodology.

REFERENCES:

[1] King, Brett., "Bank 2.0: How customer behaviour and technology will change the future of financial services", *Marshall Cavendish International Asia Pte Ltd*, 2010.

[2] Puustinen, R., "Globalisation and the Nigerian banking industry: efficiency and legitimacy considerations in the adoption of electronic banking (e-banking) services", *International Journal of Management and Decision Making*, Vol .7, No. 5, 2006. pp. 494 - 507.

[3] Z. Yin, Y. Gao and B. Chen, "On Development of Supplementary Criminal analysis System Based on CBR and Ontology", *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, Vol. 14, Oct. 2010, pp. 653-655.

[4] Browning, Dennis, and Gary C. Kessler. "Bluetooth hacking: A case study", *Proceedings of the Conference on Digital Forensics, Security and Law. Association of Digital Forensics, Security and Law*, 2009. pp. 115-127.

[5] Cheng, J., Wong, S. H., Yang, H., & Lu, S. (2007, June). "Smartsiren: virus detection and alert for smartphones", *In Proceedings of the 5th international conference on Mobile systems, applications and services, ACM*, 2007, pp. 258-271.

[6] Y. G. Jeong, Y. Y. Lee, K. R. Park, J. W. Kim, D. H. Kim, "The Study on Methods to Utilize Agile Methodology Applicable to Financial Security Software Projects", *The 2nd International Conference on Computing Convergence and Applications*, August 17-20, 2017, pp. 21-24.

[7] Mercaldo, Francesco, et al. "Ransomware steals your phone. formal methods rescue it", *International Conference on Formal Techniques for Distributed Objects, Components, and Systems. Springer, Cham*, 2016, pp. 212-221

[8] Scaife, N., Carter, H., Traynor, P., & Butler, K. R., "Cryptolock (and drop it): stopping ransomware attacks on user data",

In Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on, IEEE, 2016, pp. 303-312.

[9] Singh, Gurpreet., “A study of encryption algorithms (RSA, DES, 3DES and AES) for information security”, *International Journal of Computer Applications*, Vol. 67, No. 19, 2013, pp. 33-38.

[10] Verma, Deepika, and Er Karan Mahajan., “To Enhance Data Security in Cloud Computing using Combination of Encryption Algorithms”, *International Journal of Advances in Science and Technology (IJAST)*, Vol. 2, Issue 4, 2014, pp. 41-44.

[11] Ian Sommerville, “Software Engineering, 10th Edition”, *Pearson*, 2015, pp. 75-76, 425-428.

[12] Martin Fowler, Beck, Kent; et al., “Manifesto for Agile Software Development”, 2001.

[13] Campanelli, A. S., “A Model for Agile Method Tailoring”, *Projetos e Dissertações em Sistemas de Informação e Gestão do Conhecimento*, Vol. 3, No. 2, 2014.

[14] R. Carbon, M. Lindvall, D. Muthig, P. Costa, “Integrating product line engineering and agile methods: flexible design up-front vs. incremental design”, *1st International Workshop on Agile Product Line Engineering (APLE06)*, 2006.

[15] Pete Deemer, Gabrielle Benefield, Craig Larman, “The Scrum Primer v1.1,” *Scrum Training Institute*, 2008, pp. 4-5.

[16] Rising L, N. Janoff, “The Scrum Software Development Process for Small Teams”, *IEEE Software*, Vol. 17, No. 4, 2000, pp. 26-32.

[17] BASSIL, Youssef. “A simulation model for the waterfall software development life cycle”, *International Journal of Engineering & Technology (IJET)*, 2012, Vol. 2, No. 5, pp. 742-749.

[18] IEEE-STD-610, “A Compilation of IEEE Standard Computer Glossaries”, *IEEE Standard Computer Dictionary*, 1991.

[19] Andrew Stellman, Jennifer Greene, “Applied Software Project Management”, *O'Reilly Media*, 2005.

[20] Fisher, John, D. Koning, and A. P. Ludwigsen., “Utilizing Atlassian JIRA for Large-Scale Software Development Management”, *No. LLNL-CONF-644176. Lawrence Livermore National Laboratory (LLNL), Livermore, CA*, 2013.

[21] Schwaber, K. and M. Beedle, “Agile Software Development With Scrum”, *Upper Saddle River*, *NJ, Prentice-Hall*, 2002.

[22] Schwaber, K., “Scrum Development Process. OOPSLA'95 Workshop on Business Object Design and Implementation”, *Springer-Verlag*, 1995.

[23] Abrahamsson, Pekka, et al. “Agile software development methods: Review and analysis”, *arXiv preprint arXiv:1709.08439*, 2017.

[24] Schwaber, Ken. "Scrum development process”, *Business object design and implementation*. *Springer London*, 1997, pp. 117-134.