

A Hybrid Model Based on Agile Development

Zheng Liu

School of Computer Engineering and Science
Shanghai University
Shanghai 200444, Peoples Republic of China
jourrey@shu.edu.cn

Shardrom Johnson

Information Centre
Shanghai Municipal Education Commission
Shanghai 200003, Peoples Republic of China
jshardrom@shmec.gov.cn

Gang Wang

School of Optical-Electrical & Computer Engineering
University of Shanghai for Science & Technology
Shanghai 200093, Peoples Republic of China
rogerwg@st.usst.edu.cn

Xubin Liu

School of Optical-Electrical & Computer Engineering
University of Shanghai for Science & Technology
Shanghai 200093, Peoples Republic of China
bingo@st.usst.edu.cn

Yang Song

School of Computer Engineering and Science
Shanghai University
Shanghai 200444, Peoples Republic of China
bigsheep@shu.edu.cn

Abstract—One key factor of successful software project is to control software development cycle, reduce labor cost, improve software quality. Agile software development method was widely used in practice, the efficiency of software development was also increased in fact. However, for the bad control on development cycle and labor, the application of agile software development is facing more and more problems, and then the expected effect is more and more difficult to achieve. Analysis the problems on agile software development method, a hybrid model based on agile development is put forward as RISI(Relevance between Incremental and Story Iteration). RISI model emphasizes the fusion around story iteration and incremental. With the RISI, the process of software development can be effective control on the development schedule of story, reduce the overall impact of story. The application in the actual software development projects has achieved very good results.

Keywords—Software Development; RISI Model; Invremental; Story Iteration; Agile Software Development

I. INTRODUCTION

As the demand changes of software project become quick, function of software become abundant, architecture of software become complex, the evolution and development of software development model become complicated[1-3]. The traditional model, such as waterfall model, spiral model and rapid prototype model, have not been adapt to the current demand of the market environment[4-5]. Therefore, some new values and principles about software project development have been put forward by experts, scholars and IT practitioners[6-7].

In late 1990s, a series of agile software development methods, such as extreme programming, adaptive software development, dynamic system development, have been proposed. Being a human-center, iterative and incremental method, agile software development has the goal to improve the development efficiency and the response ability. But in the application, agile software development always degenerate into the build-and-fix model by the uncertain user demand[8-10]. It is contrary to the basic principle of agile development for developers. Therefore, a hybrid model based on agile development is put forward as RISI (Relevance between Incremental and Story Iteration) according to the various software development models, such as rapid application development model[11-12], incremental model[13], and so on.

II. THE BASIC STRUCTURE OF RISI MODEL

For the universal problems around the software development model, it is popular to combine several models to complete one software development. Based on the above ideas, RISI model is introduced to implement the step model of agile development concretely.

With the comprehensive advantages of RAD model, incremental model and agile development[6-9,14], the process segmentation and the incremental sequence generation are used in RISI model to improve the efficiency of software development. And then, the project schedule pool and the Bug state pool are used to construct the story iteration to achieve good results in practical application.

As shown in Fig. 1, RISI model is mainly composed of RAD part, incremental part and story iteration part.

- RAD Part.

Some procedures in RAD model, such as business needs, project scope, constraints, and system requirements, are quoted in RISI model to translate user needs into working models[15]. In this way, the basic structure of the software system will be simplified, convenient for the management. And then, development team can be grouped for the task completion, and developer can be completed the task of oneself.

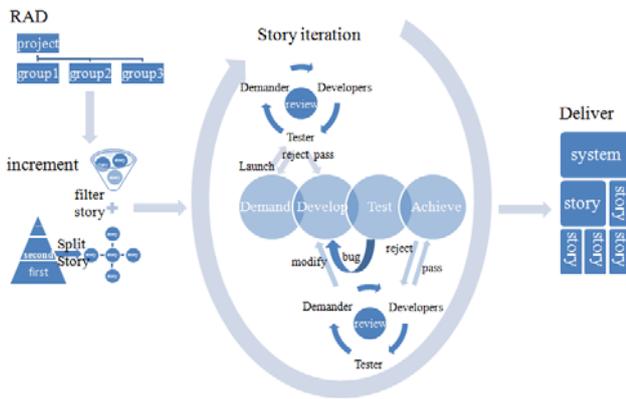


Figure 1. The structure of RISI model.

- Incremental Part.

According to the incremental model, RISI model firstly generates an incremental cycle sequence. In the sequence, the first increment is defined as the core function of software system, which realizes the basic function demand. Then, the first increment of each module will be used for requirement analysis, story split, and prototype design. Finally, the use views on current incremental will be regard as the new features and functions of the next incremental.

In incremental part, it is repeated after each incremental release, until the final perfect products. In this process, the aim of story split and prototype design will be given the priority development of important story, the risk management with the incremental model, and the basis of RISI model implementation. Therefore, the development of software system will be macroscopic order rather than random[16].

- Story Iteration Part.

By agile software development, the separated story will be iterated in the project schedule pool and the Bug state pool, so as to achieve the needful iteration result, and deliver the available different versions for user[14]. Figure 2 shows the project schedule pool and the Bug state pool.

Story is stored by the project schedule pool, which contains eight states: waiting for review, waiting for development, start development, waiting for test, start test, waiting for acceptance, start acceptance, finish.

Bug related on story is stored by the Bug state pool, which contains three states: untreated, resolved, non-existent (false positives or unable to reproduce).

III. STORY ITERATIVE PROCESS IN RISI MODEL

Process of story iteration in RISI model consists of iterative process in the project schedule pool and in the Bug state pool.

- Iterative Process in the Project Schedule Pool.

The iterative process in the project schedule pool can load the entire process of software system development, track the schedule of software system development, and come into being the progress report of software system development.

1) *Procedure one:* Iterative Process in the Project Schedule Pool.

a) *The first step:* Put the separated story into the project schedule pool, and set the status to “waiting for review”.

There are two methods to put the story into pool.

One method: firstly, perform analysis and design of the whole software system development. And then, split the story and rate the story. According to the level, the higher priority story will be used at first.

The other method: Only perform analysis and design of the higher priority story, and split the story. And then, the story will be used directly.

The advantage of “one method” is that the whole process of software system development has unified planning, and assembly of each node is achieved in the planning blueprint. But, the disadvantage of “one method” is that the steps are rigid as waterfall model, all the next step must be waiting until the demand analysis is completed. Although agile development can be performed after story split, long-term obstruction will be initiated by the state of “not agile” in this step.

Relative to “one method”, the advantage of “the other method” is the flexibility on planning. Based on the coarse structural planning of software system, the higher priority story can be analyzed, designed and split to develop as early as possible. Certainly, the disadvantage of “the other method” is that some existent design flaws will be not satisfied the following demands in module integration, the software development will be all over again.

b) *The second step:* Together with the demander, the developer and the tester, review of the separated story is in progress.

Review contents include business justification, technical feasibility, appropriate scale, and confirmed testpoint. In this step, the higher priority story will be reviewed by almost all part of the software system development. The qualified story is set the status to “waiting for development”, and the unqualified one is send back to modify in “the first step”.

c) *The third step:* According to the demand document, the story with “waiting for development” is in detailed design.

The key of detailed design is to understand the story role in the whole software system development. After detailed design, the development responsible is accountable to the story, and can decide when the story is set the status to “start development” by the priority. If the demand is not reasonable,

the review is not fully considered, and the later story detailed design is not achieved, communication with the relevant responsible will be important. According to the results of communication, the story is probably considered the new demand analysis, and set the status to “waiting for review”.

d) *The fourth step:* The tester use test case into the story with “start development”.

Test case is put forward to help the developer running self-test to improve code quality and reduce communication cost after developing. When self-test is passed, the story can be set the status to “waiting for test”.

e) *The fifth step:* The tester set up test environment.

After the test case for the story is given by the tester, the test environment is needed to build. According to the priority of story, the story can be set the status to “start test”.

f) *The sixth step:* The tester implement the test.

The story is tested in the test environment by the test case. After passing the testing, the story will be taken part in software integration to the further test. And then, the story will be set the status to “waiting for acceptance” after the further test, or a bug will be brought and given the corresponding status, high, middle and low.

When bug is set as “high”, it means that the current story should be interrupted development, and bug should be repaired immediately. When bug is set as “middle”, bug will be repaired after the current story has been development and set the status to “waiting for test”. Being the status “low”, the bug repair will be achieved before the delivery of software system, and it has nothing to do with the developer schedule.

g) *The seventh step:* The demander determine the specific arrangements of acceptance.

According to the schedule and the priority of story, the demander determines the sequence of acceptance and set the story status to “start acceptance”.

h) *The eighth step:* The demander and user implement the acceptance.

The demander and user implement the acceptance with the demand. After accepted, the story will be set the status to “finish”. If not accepted, dispatch meeting, refers to the stand meeting held approximately 10-15 minutes, will be held to discuss the reason[6]. When the existence of imperfect demand and wrong change, the story should be reviewed and implemented again.

- Iterative Process in the Bug State Pool.

The function of bug state pool is use to loading the feedback of Bug code in the software development process to quality evaluation later.

Relative to the iterative process in the project schedule pool, the iterative process in the bug state pool is simply to record the current status of Bug, and make a relevance to a corresponding story in the project schedule pool. If the developer submit the story to test, and the tester find a bug in existence, then the bug will be put into the bug state pool, and be repaired by order according to the priority.

IV. THE ADVANTAGE OF RISI MODEL

Integrated the strengths of the common software development model to complete the software development, the advantages of RISI model is embodied in the fine work and fast cycle.

A. The fine project schedule and the fine risk control

Inherited the advantages of agile development, RISI model have the fine schedule control of software system development. Compared with the general schedule by human evaluation in the past, RISI model can make the story control with the ultra fine grain, and provide the stronger data support. In RISI model, all of the story state can be clearly identified to the assignment of tasks. Therefore, even if some people have changed in the development schedule, it can only be effect on a story, not on the whole schedule.

As Figure 3 shown, when performance audit function of a system was in development, it was divided into 15 stories by the demand. Monitoring of each story can be used by the project schedule pool, and can also be used by the project risk monitor sheet.

The total number of Story		15				
KPI \ Status	Start development	Start test	Bug	Secondary bug	Finish	
Number	3	1	4	1	1	
Percent	20%	6.67%	26.67%	6.67%	6.67%	
Ratio	3	1	4	1	1	
Secondary bug rate	—	—	—	25%	—	
Online rate	—	—	—	—	53.33%	
Quality factor	—	—	—	—	0	

Figure 3. The project risk monitor sheet of RISI model.

B. The fast project development cycle

Inherited the advantages of RAD model, incremental model and agile development, RISI model has own characteristics as Figure 4 shown.

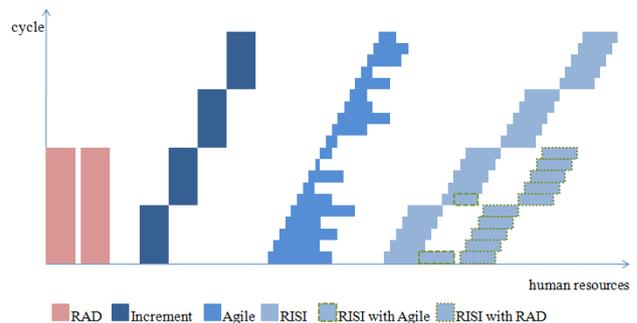


Figure 4. The characteristics of RISI model.

In RAD model, there are more need of enough sufficient human resources and prior commitments in the development. Abandoning the lack, the software system development in RISI model is divided into several equal parts with the same human, the cycle of the whole development is shorten, and

the advantage of creating a “full-featured system” in short time is retained.

In incremental model, it is easily led to build-and-fix model by the flexibility. The development of demand iteration is used in RISI model to bring about the fast delivery of software system at the same human resources.

In agile development method, the communication among the developers is emphasized; the demand planning and the document designing are ignored. Based on the advantage of the project schedule and the risk control, the development of software system is made in RISI model not only on the time driven, but also with the document designing for the latter modifications [17].

V. CONCLUSION

In general, quick response and staff communication is the main ideas of software system development, and management strategy is the further core idea to software development method[18]. As a hybrid model based on agile development, RISI model is submitted to combine quick response, staff communication and management strategy.

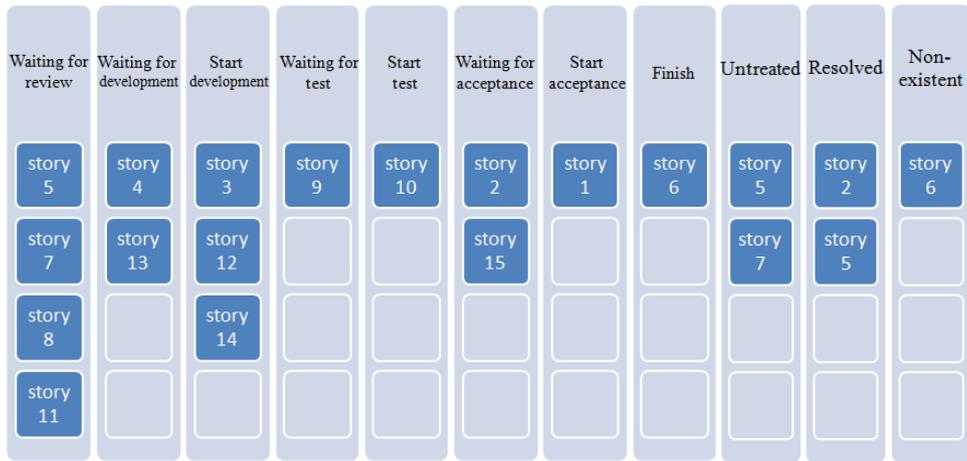
In RISI model, story is a unit to demand, design, review, development and testing. After testing, the system result will be directly seen by user in theory. In the author's practice, with the continuous development of software systems, the corresponding story will be delivered into software system by steps, and more outcomes will be appeared.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grants No.61303097, by Shanghai Key Discipline Construction Project (J50103), and by the Graduate Innovation Fund Project of Shanghai University (SHUCX070037; SHUCX120105)

REFERENCES

- [1] John A. McDermid, Keith H. Bennett. “Software Engineer Research: A Critical Appraisal”. IEE Proceedings – Software, vol. 146, no. 4, pp. 179-186, 1991.
- [2] Barry W. Boehm. “Anchoring the Software Process”. IEEE Software, vol. 13, no. 4, pp. 73-82, 1996.
- [3] Elke Pulvermüller, Gerhard Goos, Uwe Aßmann. “New Software Composition Concepts”. Science of Computer Program, vol. 56, no. 1-2, pp. 1-4, 2005.
- [4] Lung-Chun Liu, Ellis Horowitz. “A Formal Model for Software Project Management”. IEEE Transactions on Software Engineering, vol. 15, no. 10, pp.1280-1293, 1989.
- [5] Alan Mark Davis, Ann S. Zweig. “The Missing Piece of Software Development”. Journal of Systems and Software, vol. 53, no. 3, pp.205-206, 2000.
- [6] Jim Highsmith, Alistair Cockburn. “Agile Software Development: The Business of Innovation”. IEEE Computer, vol. 34, no. 9, pp. 120-122, 2001.
- [7] Christof Ebert, Pekka Abrahamsson, Nilay V. Oza. “Lean Software Development”. IEEE Software, vol. 29, no. 5, pp. 22-25, 2012.
- [8] Robert Cecil Martin. “The Test Bus Imperative: Architectures That Support Automated Acceptance Testing”. IEEE Software, vol. 22, no. 4, pp. 65-67, 2005.
- [9] Kent Beck, Barry W. Boehm. “Agility through Discipline: A Debate”. IEEE Computer, vol. 36, no. 6, pp. 44-46, 2003.
- [10] Yi Zhang, Xiaoli Liu, Weinong Wang. “Policy Lifecycle Model for Systems Management”. IT Professional, vol. 7, no. 2, pp. 50-54, 2005.
- [11] James Martin. “Rapid Application Development”. Indianapolis: Macmillan Publishing Co., Inc., 1991.
- [12] Paul Beynon-Davies, C. Carne, Hugh Machay, Douglas Tudhope. “Rapid Application Development (RAD): An Empirical Review”. European Journal of Information Systems, vol. 8, no. 3, pp. 211-223, 1999.
- [13] Craig Larman, Victor R. Basili. “Iterative and Incremental Development: A Brief History”. IEEE Computer, vol. 36, no. 6, pp. 47-56, 2003.
- [14] Mike Cohn, Doris Ford. “Introducing an Agile Process to an Organization”. IEEE Computer, vol. 36, no. 6, pp. 74-78, 2003.
- [15] George Armitage Miller. “The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information”. The Psychological Review, no. 63, pp. 81-97, 1956.
- [16] Martin Fowler. “Public versus Published Interfaces”. IEEE Software, vol. 19, no. 2, pp. 18-19, 2002.
- [17] Alan Mark Davis, Pradip Sitaram. “A Concurrent Process Model of Software Development”. ACM SIGSOFT Software Engineering Notes, vol. 19, no. 2, pp. 38-51, 1994.
- [18] Jiaoxiong Xia, Jue Gao, Yi Tang. “Demand Driven and Its Development”, Computer Engineering, vol. 24, no. 5, pp. 49-51, 62, 1998.



The project schedule pool

The Bug state pool

Figure 2. The project schedule pool and the Bug state pool of RISI model.