Research Process on Software Development Model

Jiujiu Yu
College of Computer Engineering, Anhui SanLian University, Hefei 230601, China
yjjyjL@163.com

Abstract. Software development model is the form of the organization and management of software development which affects the quality of software development directly, and it is one of the most important research contents of software engineering too. Firstly, we analyze and summarize a series of software development models for total thirteen types which are applied in software development generally, and survey the characteristics of each model. Secondly, some of the improved software development models and their applications according to certain application fields are introduced briefly. Finally, as the future research work, both software development based on 4GT (The Fourth Generation Technology) and software intelligent development based on data-driven in big data era are prospected in this paper.

1. Introduction
Software development model is an expression of the whole process of software development, which defines the main activities clearly to instruct the working tasks on software development that to be completed. It is the structural framework of all the work and tasks which are implemented by system development, running and maintenance to across the entire software life cycle, and the relationship between the various phases of software development activities are also given [1]. As we all know, there are various software development approaches that defined and designed which are used or employed during development process of software [2]. These approaches are also referred to software development models. However, each model follows a particular life cycle to ensure success in process of development, and its advantage and disadvantage are all existed indeed. Software development cycle and software quality will be affected by software development model directly, and it is the important form of the organization and management for software projects in modern IT enterprise.

2. Software Development Model

2.1. Waterfall Model
Waterfall model is a traditional software development model, but the framework of the model is much simple. The model consists of development phases of requirements analysis, designing, coding, testing, and maintenance, but each phase must be executed in queue, and requirements must be known up front. But, it's difficult to imagine every detail in advance in reality. Most projects start out with some uncertainty, and more details are learned as the project progresses. The major disadvantage of model is that problems in system aren't be discovered until late in process(testing phase), but it leaves little
time for correction, resulting in potentially disastrous which effects on project schedule and cost [3]. The application of this model in modern IT enterprise is little now.

2.2. Prototype Model
Prototype model is another simple software development model, but is widely used now. The core idea of prototype model is that the developers and users quickly determine the most important requirements for users at first, and a prototype of a software product can be achieved rapidly. According to the evaluation and feedback by users, the prototype is modified constantly by developers until users’ satisfaction. Prototype model is suitable for environment which the requirements for users are uncertainty at the beginning of development. Developers should have good experience on development, so the most important requirements could be achieved by establishing prototype of software product [4]. However, applications on this model may cause the development process immobilized, the innovative ideas for developers will be limited, and it is difficult to update and manage the documents.

2.3. Incremental Model
Incremental model (shown as Figure 1) is a non-monolithic model. The model combines the sequential features on waterfall model with the rapid iterative features on prototype model [5]. Based on the previous increment, the latter increment is developed, and each increment can be developed by using a waterfall or a rapid prototype model.

A software product can be delivered which are divided into several periods, it allows users to use some of the basic functions, and other functions of the product will be continued to be developed by developers at the same time. The model is also suitable for software projects with design risks and the uncertain requirements in early time.

2.4. Spiral Model
Spiral model (shown as Figure 2) uses a periodic method to develop the system. Each cycle includes four phases: requirements definition, risk analysis, project implementation and evaluation. The four phases are iterated. Software development is promoted to a level when an iteration process is finished in the model. The advantage of the model is that risk analysis is emphasized for developers and users in each round of iteration, so the model is suitable for the development of large system. But, the disadvantage of the model is that the application of spiral model requires an amount of experts with experience on risk assessment to participate, and excessive iterations will increase the costs on development and cause submission delaying.

2.5. Fountain Model
Fountain model (shown as Figure 3) is based on user driven and object driven, and it is suitable for object-oriented software development process.

There are no specific orders and clear boundaries for each phase of development in the model, and all phases of development can be taken place international. The fountain model has the advantage that a progressive piece of software is allowed to be added on each iterative development, and easier to supple the missing in other phases of development at any time during one development phase. The disadvantage of the application of fountain model is that in the iterative development may face the possibility of adding a variety of information, requirements and information at the time, it requires a large number of developers, and is not convenient to project management.
2.6. XP Model

XP (Extreme Programming) model (shown as Figure 4) is a lightweight and progressive model which based on software agile development method.

In XP model, the whole development process is divided into six phases and is shown as figure 6. The core ideas of the model are a set of values, principles and practices for rapidly developing high-quality software that provides the highest value for the customer in the fastest way possible [6]. XP model is widely used on development of small software projects with high risk and has the advantage of adapting to rapid changes in user requirements in modern IT enterprise. Applications on XP model enables developers to focus on coding and avoiding unnecessary to work of documents management.

2.7. RUP Model

RUP (Rational Unified Process) model (shown as Figure 5) describes the life cycle of an entire software development over time by the development of managing the project cycle [7]. RUP model has nine core workflows which are used in turn, and they are emphasized repeatedly on each iteration. Architecture based on centered, use cases driven, iterative and incremental are the main features on RUP. However, due to the complexity of the development process in the RUP model and its close links, it is not suitable for small software development projects. At present, many software developers try to cut or simplify some workflows of RUP model appropriately and combine them with the X model to apply on development of small software projects successfully.

2.8. RAD Model

RAD (Rapid Application Development) model (shown as Figure 6) is a variation of waterfall model that completes the rapid development of software by using of reusable components [8].

(*Note: BM-Business Modeling, DM-Data Modeling, PM-Process Modeling, AG-Application Generation, TR-Test and Repetition)

RAD model is actually an incremental development model with process of business modeling, data modeling, process modeling, application generation, test and repetition which emphasizes extremely
short development cycles that allow a development team to create a fully functional system in a short
of the time, such as 60 to 90 days. By the way, Development processes in RAD model are not to use
the traditional third-generation programming language to create software, that is to say, RAD model is
not suitable for development which the system is not reasonably modularized, and is not suitable for
the system with higher risk on requirements, or the interfaces on components are need to be changed
frequently either.

Figure 5. RUP model.

Figure 6. RAD model*.

2.9. WINWIN Model
WINWIN model (shown as Figure 7) combines characteristics on spiral model with prototype model,
emphasizing risk analysis and identification, and through earlier negotiations to achieve a double
winning result for users and developers on software development activities [1]. Three milestones of
goal on life cycle (Defines a set of goals for each major software engineering activity.), architecture on
life cycle (Establishes the goals that must be met when system and software architectures are defined.)
and Initial operation ability (Represents a set of goals, as well as the preparation for these goals and
pre-installation sites for the software to be installed or sold which is associated with the helps for
software.) are in WINWIN model, to represents the three different views on progress of the project
development for spiral.

The advantage of WINWIN model is both users and developers are all understand the risk which is
emerged at each level of evolution and are responded accordingly [9]. They will get a balance on
developing resources. On one hand, most of the functions of system can be achieved for users. On the
other hand, developers could also gain the achievable budgets and frames of time in reality.

2.10. Intelligent Model
Intelligent model (shown as Figure 8) is also known as a "knowledge-based software development
model", which combines a waterfall model with an expert system to integrate knowledge system
(based on software engineering knowledge) and expert system (including knowledge rules in the
application area), using expert system to help software developers to work. The intelligent model has
a set of tools (such as data query, report generation, data processing, screen definition, code generation,
high-level graphics functions and spreadsheets, etc.), each tool allows developers to define certain
characteristics of the software at a high level and automatically generate the developer-defined
software as source code [10].

But, it requires the fourth generations of language (4GL) to support. At present, intelligent model is
still based on framework of waterfall model, but prototype method and object-oriented development
technology are applied in different phases of intelligent model [1]. However, intelligent model is
entirely depended on theoretical knowledge. It needs to establish a knowledge base through
knowledge acquisition and knowledge representation. It also needs cooperation from experts in
specific fields and is generally applied to the development of complex expert decision systems.
2.11. Parallel Model

Parallel model focusing on the state of a series of technical tasks and activities in software engineering, and these activities are defined as an activity network.

Each activity on the web can be occurred on parallel with other activities, and this model provides an accurate view of the current state of the project. For example, an activity diagram on software process of parallel development model is shown as Figure 9. Parallel model is not driven by time, but by requirements for users, management decisions, and result reviews, enabling parallelization of the entire process of software development [1]. The application of the model is more suitable for system development with C/S (Client/Server) structure, because the C/S application is implemented by multiple components, and each component can be designed and implemented in Parallel.

2.12. Component-based Model

The development processes in component-based model are iterative, and usually five phases are consisted in the model as it is shown in Figure 10. Components have been rapidly developed as important software technologies and tools, such as EJB (SUN), DCOM (Microsoft), CORBA (OMG), etc. These are some typical new technologies and tools which are widely used in component-based development. The core of the application of the model is to start with the development activity from identifying the candidate component by searching the existing component base in order to confirm whether the required component is already existed or not [1] [11]. If the component is already existed, it is reused that from the component base. If it does not exist, the component is needed to be developed by using an object-oriented method.

The application of the model needs to modularize the system which is to be developed, and reuse one or more software components in component base by the supporting of a certain component model [11]. An application on software system is constructed by developers through a corresponding combination method. Now, many software companies start to design and use component assembly and interface specification standards customized to carry out software development activities, but the component interface standards need to be further generalized. Besides, there are many software experts who have proposed many efficient new component assembly algorithms to complete large-scale software system development activities. The author for the paper suggests the readers who are interested in this could read those references from website.

2.13. Architecture-based Model

Software architecture is the core of this model, which based on the component-based development method. The process of analysis and design are based on iterative incremental approach, mapping the functional design space to the structural design space and then maps the structural design space to the system design space again[1, 2] [12]. The architecture-based model is shown in Figure 11.

Acquisition and analysis of requirements which based on architecture first, to provide better support for the transition from the analysis phase to the design phase. And then based on the results of the requirements analysis, the design of the architecture is performed. When designing of the
architecture, it is the same as the component-based development model to search matching components according to the syntax and semantics of the constituent elements. Software component reuse has been promoted from code-level reuse to architecture-level reuse. Of course, the entire development process presents iterations. The advantages of this model make the framework of the software system more clear and conducive for designing, development and maintenance of the system [1].

2.14. Software Development Models in Certain Application Fields
Recently, many software developers and researchers are devoted to making appropriate improvements on existing development models according to the characteristics of actual software projects for certain application fields, redesigning the processes or activities for new development models, and get good results. For example, improved development model of RUP for instructional software [13], development model of multi-media instructional software based on CSCW(Computer Support Cooperative Work) [14], inverted A-model for stable software development [15], Three Triple iterative model based on agile [16], interlayer model for software development based on collective intelligence [17], software development cooperative model for large scale project [18], double iteration model of agile software development [19], component and aspect-based MDA(Model Driven Architecture) software development model [20], etc. But, some in-depth researches such as the constituent elements, characteristics of applications, application environment, and the analysis on advantages or disadvantages of these models which are listed above, readers who are interested in those can search for the references from the website for better learning.

Figure 9. Parallel model. Figure 10. Component-based model. Figure 11. Architecture-based model.

3. Summary on Software Development Model
Software development model has been evolved along with the changes by development ideas on software and undergone the mainly following stages.

Firstly, the idea of structured programming is used as guidance to software development model, such as waterfall model. In the model, various development activities are carried out must be one by one, emphasizing the sequence in stages of development. If there are obstructions in the development process, the development efficiency will be affected seriously. Secondly, progressive development models such as prototype models, spiral models, incremental models, fountain models, RUP models, XP models, etc. Models of this type are designed driven by object-oriented thinking and they are applicable to development scenarios where the requirements for users are not clear at the beginning of the development, or the user can only provide fewer requirements at the initial phase. The various activities of the software development process are iterative, through the iterative process to achieve the gradual evolution of software functions, and eventually satisfied with software products by users [1]. Due to incomplete requirements, adopting this type of models will bring difficulties to the overall design of the software, weaken the integrity of the software design, and require the developers to have more experience on software development capabilities and risk management. Thirdly, development models which the design ideas on component development, such as component-based development
model, architecture-based model, and other improved development models for certain application fields which are mentioned above. The overall structure of the software is designed by using the results on requirements analysis, and the software system is constructed by the component-based assembly method. It is conducive to the design, development and maintenance on software structures. Of course, when the new idea on software design is put forward by software experts, it will emerge the new models for software development to improve the quality of software development soon.

4. Future Research

4.1. Software Development Based on 4GT

Software development based on 4GT (The Fourth Generation Technology) is to be an important development method of software engineering in the future. Certain features of the software are needed to be regulated at a higher level by developers. Using software tools, a program is constructed according to the developer’s specifications, and the source code is automatically generated too [1]. The method on software development based on 4GT requires the ability of software specifications to use a special language form or some graphical symbols that describes the problem to be solved by using terms that the users can be understood.

In fact, software development based on 4GT is also started from the collection of requirements, but it is unrealistic for users to specify the software requirements fully and convert it into an operable software product prototype directly. Many experts still believe that software development based on 4GT will be appeared with ambiguous for specifying in the known situation. Prof. Y.S. Zhang (Hunan Normal University) believes that if 4GT is applied in reality, it needs to be thoroughly tested first and all integration activities that are also required in other software development models must be completed. Finally, using the software product which is developed by 4GT, in the process of its construction, the effectiveness of the source code generated by software tool, as well as the maintainability of the software, etc., should be further studied.

4.2. Software intelligent development based on data-driven

Software intelligent development has become one of the most important research trends in software engineering in big data era. Intelligent Development Environment (IntelliDE) and Software Knowledge Graph (SKG) are put forward in [21] to provide intelligent assistance in the life cycle of software development and get some progress. However, modern software development activities have been formed a resource of big data. Intelligent software tools can be provided to developers with recommendation and intelligent searching services which are based on data knowledge, thereby forming a software-intelligence integration of “Person-Tool-Data” for development method, technical system, environment, and etc.

Therefore, in the process of intelligent software development based on data-driven, research fields including the collection and aggregation of software engineering data, the extraction of knowledge for big data of software engineering, the innovation exploration of new methods, technologies, and tools for intelligent release in different stages of software testing, software verification, and software maintenance will be the challenging research in the future.

Acknowledgments

Corresponding author (Jiujiu Yu) of this paper acknowledges financial support from the projects of "Software Engineering-Excellent Resource Sharing Course of Anhui Province" (No.2016gxxk048), "Research on Software Exploratory Testing and Its Key Technology" from Anhui SanLian University (No.KJZD2017008), and "Teaching Team on Curriculum of Software Engineering" from Anhui SanLian University (No.15zlgc029). Finally, I would like to express my heartfelt gratitude to all authors of the references which are listed at the end of this paper.
References
[1] Y.S.Zhang, X. Li, Software Development Models: a Survey, Journal of Computer Engineering and Applications, issue 3, pp.109-110, 2006.
[2] J.J.Yu, A Short Course in Software Engineering, Tsinghua University Press, 2015.
[3] Information on https://www.cnblogs.com/wintersun/p/6828400.html, 2017.
[4] S.M. Zhu, Software Testing (Second Version). Posts & Telecom Press, 2016.
[5] X.H. Liu, Software Engineering and Project Management, Peking University Press, 2009.
[6] S.J. Wu, Improvement on Quality Testing Project of SI Software Based on Agile Scrum Mode, MASTER'S THEESIS of Southwest Petroleum University, 2015.
[7] Y.M. Du, S.X. Li, Estimation Process Model for RUP Project, Journal of Computer Science, vol. 40, issue 6, pp.21-26, 2013.
[8] J.X. Xia, Z. Liu, X.B. Liu, Y.Song and J.J.Yuan, Incremental Story Iteration Model Based on Rapid Application Development, Journal of University of Shanghai For Science and Technology, issue 6, pp.578-583, 2014.
[9] R.M. Zhang, D. Yang and J. Li, Design of requirements negotiation tool based on WinWin theory, Journal of Computer Engineering and Applications, issue 1, pp.100-104, 2009.
[10] Michael Negnevitsky, Artificial Intelligence: A Guide to Intelligent Systems (Third Version), China Machine Press, 2012.
[11] X.Zou, Method of Construction: Modern Software Engineering, Posts & Telecom Press, 2018.
[12] Z. Luo, S. Q.Yuan, J. L.Yuan and L. Li, Software Engineering. Posts & Telecom Press, 2017.
[13] J.J. Yu, Research on designing and achievement on RUP improvement model of instructional software, Journal of E-education Research, issue 4, pp.76-81, 2012.
[14] Z.L. Gù, Research on development model of multi-media teaching software base on CSCW, Journal of Computer Engineering and Applications, issue 9, pp.1628-1630, 2006.
[15] S.J. Chen, Inverted A-Model for Stable Software Development, Journal of Software, vol. 37, issue 12, pp.07-11, 2016.
[16] J.J. Ma, Three Triple Iterative Model Based on Agile, Journal of Electronic Technology & Software Engineering, issue 6, pp.52-54, 2017.
[17] B.Lu, L.Liu, J.R. Li and L.M.Jiang, Artificial Intelligence and Its Application, Tsinghua University Press, 2017.
[18] Y.L. Si and W.H. Liu, Research of a New Software Development Cooperative Model, Journal of Microelectronics & Computer, issue 5, pp.73-76, 2012.
[19] D.Q. Xie, Double Iteration Model of Agile Software Development, Journal of Computer Applications and Software, vol. 29, issue 6, pp.176-178, 2012.
[20] Y. Wu, J.Y. Qian and Y. Liu, MDA-Based Component Employer Method Research and Implementation, Journal of Control & Automation, issue 27, pp.198-200, 2010.
[21] Z.Q. Lin, B. Xie and Y.Z. Zou, Intelligent Development Environment and Software Knowledge Graph, Journal of Computer Science and Technology, vol. 32, issue 2, pp.242-249, 2017.