Software Development Technology Initiatives at NTT Laboratories

Takashi Hoshino, Kazuhito Suguri, Katsuyuki Natsukawa, and Yasuharu Yamada

Abstract

A large amount of software is used within the NTT Group of companies. Development is in progress on much of this software, but this development does not always run smoothly. Because of this, the NTT Software Innovation Center conducts research and development of software development technology in order to increase the stability and efficiency of software development. This article reports on some of the research and development underway.

Keywords: software development, software engineering, development processes

1. Introduction

A survey conducted by the Information-technology Promotion Agency (IPA) indicated that only about 15% of software projects exhibit no problems after release, and more than 10% of projects have more than ten trouble issues. The three key aspects of software development are quality, cost, and delivery (QCD); from a developer's perspective, delivery dates are met in about 85% of cases, but quality and cost targets are achieved in less than 80% of cases. From a user's perspective, cost objectives are met in approximately 80% of cases, but quality and delivery objectives are only met in approximately 70 to 75% of cases. There has been some gradual improvement when looking at changes year-to-year, but the current state is still inadequate [1].

Much software is also developed and used within the NTT Group companies. We are continually making improvements in these development projects, and efficiency is increasing, but this does not mean that all software development proceeds smoothly and steadily. In order to provide services quickly, development of the software supporting those services must proceed even more efficiently.

The NTT Software Innovation Center (SIC) is conducting research and development to create software development technologies that will bring greater sta-bility and efficiency to all software development at the NTT Group companies.

2. Software development technology

Software engineering includes studying how to improve software development efficiency. The IEEE (Institute of Electrical and Electronics Engineers) Computer Society defines software engineering as: "(a) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (b) The study of approaches, as in (a)". This definition suggests an academic approach such as in establishing systematic methods. In contrast, our project is called the Software Development Technology Project and does not use the term software engineering. While the term software development technology certainly includes the systematic and quantitative initiatives of software engineering, it also implies the active pursuit of initiatives related to practical technologies commonly in use in the development workplace, which are not necessarily covered by this definition, for example, knowledge and know-how tools and other work-place aids [2].
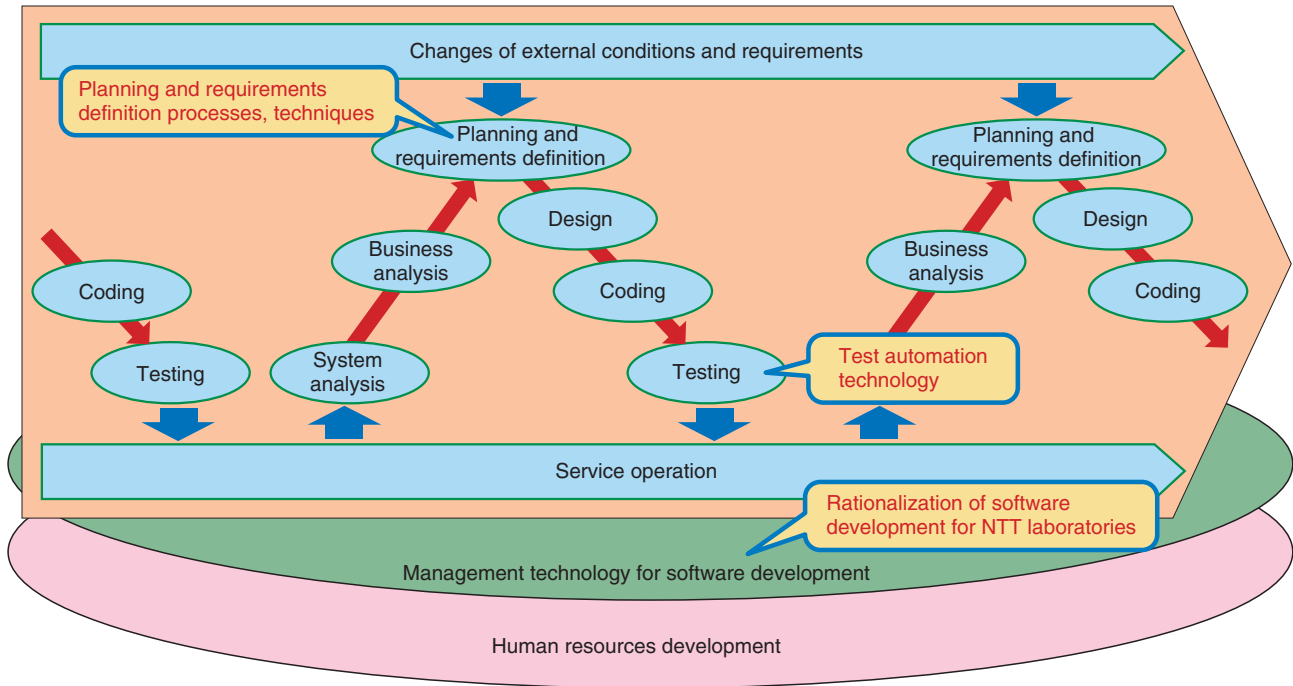
Fig. 1. Initiatives of Software Development Technology Project.

3. Research and development towards rationalized QCD

Our goal at SIC is to create software development technologies for providing services quickly, inexpensively, and with good quality. To achieve rapid service provision and to contribute to reducing the total cost of ownership (TCO), we are working on creating software development technology for application software in particular.

The typical software development process flow and activities supporting it are shown in **Fig. 1**. Before software is introduced for a service, it is developed in a series of processes, which are: *planning and requirements definition*, in which the functionality and performance required by users is decided based on objectives, return on investment, and analysis of the current system; *design*, in which the functionality and performance of the system to be provided to users is decided based on the requirements, along with how it will specifically be implemented; *coding*, in which the program is coded according to the design; and *testing*, in which the implemented program is checked

to make sure it operates correctly according to the design and requirements and that it satisfies the original user requirements. When additional development is done to improve functionality or for other reasons, this process is repeated.

These processes must be performed reliably in order to achieve appropriate levels of QCD in software development. The planning and requirements definition is the most fundamental process; it is an important elemental step in which developers ask what needs to be done or what needs to be created, and it has a significant influence on later processes. If this definition is ambiguous, a lot of work may have to be repeated, which affects all aspects of QCD: cost increases (C), delayed completion (D), and lower than expected functionality or quality (Q). Thus, it is extremely important to conduct the planning and requirements definition accurately. At SIC, we are conducting research and development (R&D) on processes for performing the planning and requirements definition, and on notation practices for documenting requirements.

The testing process involves checking the operation

of the software and increasing functional quality, so this work really focuses on ensuring quality (Q) and is the last measure to prevent poor quality. Testing also accounts for a large share of the software development work and cost, so improving testing efficiency can reduce the cost and shorten the development time, two factors that are directly related to costs (C) and delivery (D).

SIC is also developing technologies to support test planning and execution, with the goal of performing this test work more accurately and efficiently. The work done in all the steps of the planning and requirements definition and testing processes discussed here can have a significant effect on QCD. Further, each of the aspects of QCD is intimately related to each other. For example, pursuing a higher level of quality (Q) than required can incur more work, higher costs, and a longer time to ensure the higher quality level, which will have undesired effects on C and D. It is therefore important to proceed with an appropriate QCD balance, so it is necessary to manage QCD appropriately. SIC is conducting R&D on management of software development, with a view toward providing such QCD management. This includes issues such as establishing methods to evaluate quality and productivity, gathering development data needed for such evaluations, and establishing and operating software development standards that will formalize and enable the entire development process to proceed efficiently. Addressing the management of development quantitatively in this way will bring stability to each process and contribute to the overall stability of software development.

In these Feature Articles, we introduce some of SIC's initiatives related to important technologies and activities for increasing the stability and efficiency of software development, as discussed here. SIC has not performed this technical development from scratch on its own, but has used technologies that NTT Group companies already possess, as well as open source software and commercial technologies, with the aim of advancing open innovation and producing results quickly.

4. Linking practical needs and NTT Group companies

For software development technology, it is important to produce technologies that are immediately useful in the development workplace. In order to understand the needs of the software development workplace accurately, we are strengthening our ties with NTT Group companies.

Within the NTT Group, there are companies on both sides of software development: those that place orders requesting software development and then use the completed software, and those that take such orders and develop the software. We are striving for optimal technology development that will substantially increase development efficiency by listening to both sides, understanding, analyzing, and optimizing the real issues and needs in both workplaces, and advancing our work while increasing collaboration with companies that handle a lot of software development—even those just within the NTT Group such as NTT DATA and NTT Comware.

5. Towards the future

SIC has pursued these initiatives while maintaining a strong awareness of, and cooperating more deeply with, the development workplace. We will continue to work on creating technologies that are immediately useful in the development workplace and will select research themes suited to the needs of this workplace while striving to increase the stability and efficiency of software development within the NTT Group.

References

[1] The Information-technology Promotion Agency, Japan, "2012 Survey of the State of the Software Industry," Apr. 2013 (in Japanese).
[2] H. Ogawa, "Software Development and Software Engineering in Industry," Computer Software, Vol. 28, No. 3, pp. 2–11, 2011 (in Japanese).

Takashi Hoshino
Project Manager, Software Engineering Project, NTT Software Innovation Center.
He received the B.E. and M.E. degrees from the University of Electro-Communications, Tokyo, in 1989 and 1991, respectively. He joined NTT in 1991. His current research interests include software engineering.

Katsuyuki Natsukawa
Senior Research Engineer, Supervisor, Group Leader, Software Engineering Project, NTT Software Innovation Center.
He received the M.E. degree in engineering from NARA Institute of Science and Technology in 1996. He joined NTT in 1996. During 2009–2012, he worked for NTT DATA. His current research interests include software engineering.

Kazuhito Suguri
Senior Research Engineer, Supervisor, Software Engineering Project, NTT Software Innovation Center.
He received the B.E. and M.E. degrees from Toyohashi University of Technology, Aichi, in 1990 and 1992, respectively. Since joining NTT in 1992, he has contributed to the development of video-codecs, communication equipment, and content delivery platforms. His current research interests include software development processes, model-driven-development, and software metrics. He is a member of the Information Processing Society of Japan and IEEE.

Yasuharu Yamada
Senior Research Engineer, Software Engineering Project, NTT Software Innovation Center.
He received the B.E. and M.E. degrees in electronic-mechanical engineering from Nagoya University, Aichi, in 1992 and 1994, respectively. He joined NTT in 1994. His current research interests include software engineering.