

Project Propulsion Platform: P3

● Tomonori Yamashita ● Kimio Saito ● Masaru Kameda

As technologies supporting the information technology (IT) society continue to advance on an almost daily basis, demands for greater diversity and higher performance in functions are driving up the level and scale of design technologies used in IT device development. Compared to the situation just a few years ago, an extensive number of requirements must now be examined, creating an increasingly complicated development environment. Nowadays, many developers are needed to achieve early development and shipment as demanded by the market, but such a high number of project participants can increase the cost of information and knowledge sharing, thereby increasing the risk of project delays. We have constructed an environment that visualizes development status, problems, and countermeasures within a project, applies the know-how accumulated through development work, and supports improvements in development efficiency and productivity. This paper introduces the Project Propulsion Platform (P3), which facilitates the sharing of information and knowledge among developers. It reports the results obtained from applying this platform to actual design and development projects and discusses remaining issues and future plans.

1. Introduction

The development of information technology (IT) devices is becoming increasingly difficult as technologies progress and customer needs diversify. The number of transistors on a computer chip continues to increase in accordance with Moore's law, resulting in an increasingly complex product. This means that issues that must be addressed at the time of product development such as techniques for shrinking wiring dimensions and measures for lowering power consumption are likewise increasing in number. What's more, if we turn to communication devices, the functions provided by mobile devices continue to expand beyond the telephone to include Internet access, music, games, photos and videos, and e-money (Figure 1).

In such a business environment, IT companies need to develop and ship products

even more quickly than before, necessitating improvements in development efficiency and productivity.

1) Improvements in development efficiency

Despite the fact that product functions and performance are now undergoing major changes, the market demands that products be shipped as quickly as or even more quickly than before. According to International Business Strategies, Inc.,¹⁾ a consulting firm for the electronics industry, the relationship between sales for the first product to make it to market and the sales of subsequently introduced products is described as follows. The first product to make it to market will of course have 100% of the market while the ones that make it to market three months, six months, and one year later will capture only 73%, 53%, and 9% of the market, respectively. In a product market with rapidly

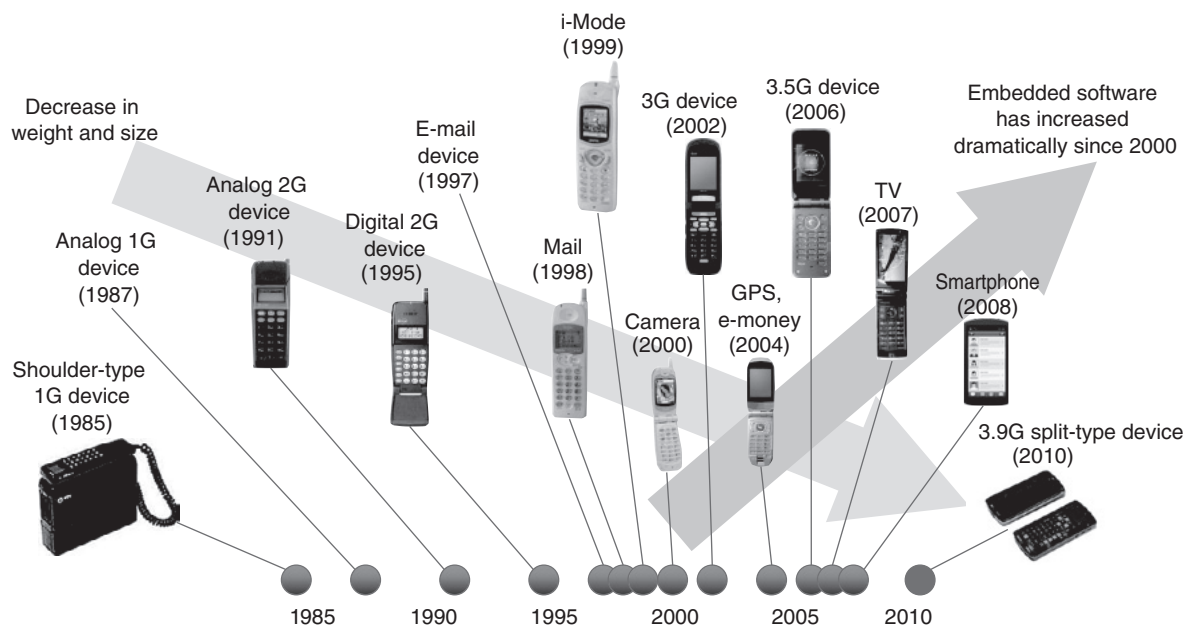


Figure 1
Advancements in mobile devices.

progressing technology and severe competition, it is imperative that companies develop their products and get them to market as soon as possible if they are to become a dominant leader in that market. This makes efficient development a matter of great urgency.

2) Improvements in development productivity

Productivity in product development is not keeping up with the pace of improvements in product scale and performance. According to a survey of various companies regarding problems in development conducted by the Japan Electronics and Information Technology Industries Association (JEITA) in 2000,²⁾ LSI developers attributed more than half of development delays to “errors in conveying specifications between developers,” “misunderstandings of complicated development technology,” and “misinterpretations of vague expressions in specifications.” According to a survey conducted by the SEMATECH (SEmiconductor MANufacturing TECHnology) consortium, semiconductor scale had been growing at an annual rate of 58% while

developer productivity had been increasing at an annual rate of only 21%.³⁾ Furuyama et al.⁴⁾ investigated the relationship between the scale of software and quality, cost, and delivery (QCD) and found that as the scale of software development increases so does the rate of quality deterioration, cost overruns, and delivery delays. At EDA Tech Forum 2009, Mentor Graphics in the United States reported on a survey that showed that development delays occurred in 66% of LSI development projects. The above data demonstrates the need for improving productivity.

This paper introduces the Project Propulsion Platform (P3), which was developed by Fujitsu for improving development efficiency and productivity.

2. Design and development knowledge

Developers use various types of information and knowledge during the development process. LSI development now requires knowledge not generally needed prior to 2000.⁵⁾ For example,

developers must now be knowledgeable of device-specific circuits, intellectual property (IP) macros, restrictions, and embedded processors. Likewise, in software development, the pursuit of higher processing speeds and safer and more secure processes as well as the diversification of user preferences is increasing the need for knowledge on achieving true real-time characteristics, advanced security measures, and compelling graphical user interfaces (GUIs).

2.1 Development process (flow)

Now, let's check the debate on development flow as a key example of design and development knowledge. Yamada⁶⁾ defined software development flow as a work process that converts problem-solving knowledge in a certain problem area to executable knowledge. This definition is also applicable to hardware development. In other words, we can think of the development process as work that converts someone's "problem-solving knowledge" into "something" that the eye can see. This conversion process has been well researched as a development model. A good example is Royce's "waterfall model,"⁷⁾ which compares the development process to the downward flow of water. There is also the "spiral" development model presented by Bohem⁸⁾ in which a cycle of analysis, design, prototyping, and testing is repeated a number of times from the initial planning stage to development completion to allow for more flexibility in the development process.

2.2 International standards

Many international standards have been established to improve work procedures and product quality. The ISO9001 standard,⁹⁾ for example, which aims to improve the quality of the development process in a spiral manner by repeating a plan-do-check-act (PDCA) cycle, has been introduced and practically applied at many manufacturing worksites. There are also organizations that measure their own maturity

in product development using the Capability Maturity Model Integration (CMMI)¹⁰⁾ approach developed by Carnegie Mellon University. The PMBOK Guide,¹¹⁾ a compilation of project management knowledge, has also found widespread use.

3. Problems and countermeasures in design and development

Companies are introducing advanced development technologies, techniques, and tools with the aim of raising development efficiency at product development sites, and advances in network technologies are facilitating concurrent development among geographically dispersed sites. Offshore development with the aim of controlling development costs is also being actively pursued. Yet, the survey conducted by JEITA in 2000 found that problems still existed in conveying and being aware of development-related information and that the formalization and sharing of knowledge (design information) created within development departments was still insufficient. As recently as 2009, delays in development projects were still numerous, suggesting that this problem had yet to be solved.

An in-house survey of development conditions uncovered a number of problems. One is that designers are increasingly designing things in a piecemeal manner due to tighter development deadlines and greater development complexity. Moreover, as this is being done in a somewhat ambiguous state in which specifications have not been firmly established due to the poor definition of user needs, development changes are increasing, thereby increasing the passing on of development-related information. At the same time, change-related information is not always adequately conveyed when a development change occurs, resulting in not a few cases in which rework must be performed. Moreover, in addition to problems associated with the transfer of development-

related information, inefficient design also results from inadequate sharing of information on an ever-changing development process affected by changes in technology and past development projects. This can add to difficulties during testing since inefficient design can result in apparently never-ending problem detection. In addition, the fast pace of technological change makes it difficult to formalize development know-how in step with those changes. As a result, there has been a trend towards the use of implicit knowledge and a skewed distribution of specific knowledge among specific developers. Interfacing by e-mail has, of course, broadened communication channels, but information overload can sometimes make it difficult to find important and essential information, creating a problem in which appropriate information fails to be conveyed at an appropriate time.

The survey also revealed that many files on the servers used by a certain department to manage development data and information had similar names, making it difficult to quickly ascertain what data was correct and who created what data. This means that a developer may not know where data created by another developer was being stored. Storing a huge amount of information in such an unorganized manner was found to be a cause of many problems.

Close examination of these survey findings shows that the key to improving development productivity when many developers are involved is to effectively and efficiently share and use the information and knowledge that flies back and forth between developers, unrestricted by space or time. With this in mind, we set out to construct a system, the P3, that could visualize the flow and status of the development process and effectively and efficiently manage the information and knowledge that need to be shared. In particular, it would

- Present development information in parallel with the development flow.
- Correctly pass development information

between processes.

- Manage the configuration and version number of development information.
- Record and reuse development flows as best practices.
- Appropriately record unexpected events and track their status.
- Record and apply atypical and transient information such as development know-how.
- Enable project members to reference the information and knowledge they need whenever they need it.
- Ensure security.
- Support access and updating from any location via a device connected to the network.

4. Visualization system

The P3 visualization system we constructed has the features and functions described below.

1) Role-based member management

This function groups the members of a development project in accordance with their roles. For each member, it controls access to various types of information on the basis of that member's role.

2) Development-process management that can make associations between deliverables

This feature has a function for controlling and passing on the deliverables from the various development processes. Once a deliverable has been registered as "output," the processes scheduled to receive it are notified. Subsequent processes can also check on the status and version number of a specific deliverable (**Figure 2**).

3) Notification of deliverables from previous processes

A development process is assigned a status such as "in-execution" and "completed," and a deliverable with a status other than "in-execution" cannot be registered. Since a subsequent process receiving a deliverable will be notified of a status change in that deliverable

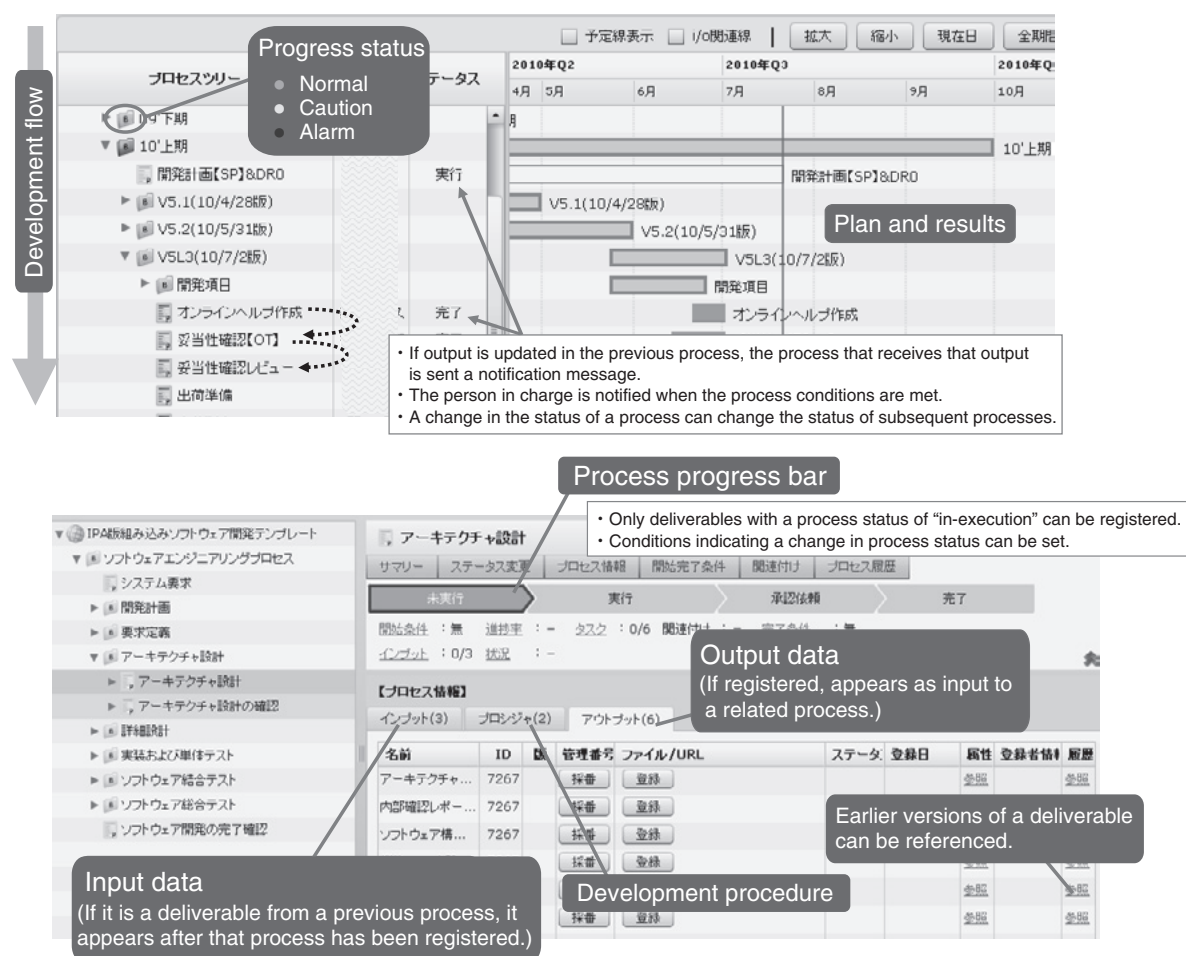


Figure 2
Development-process management screenshot (above: Gantt mode; below: detailed mode).

caused by a re-registration, any updates made to the deliverable in the previous process are delivered to the subsequent process.

4) Prevention of missed changes in previous processes

A subsequent process with a status of "completed" will be changed to "in-execution" if the status of a previous process changes to "in-execution" (status synchronization). This feature prevents a subsequent process from missing a change in a previous process.

5) Improvement in development flow

A previously defined development flow can be reused, and a development flow having particularly good results can be used within the organization as a best practice. This feature is

expected to improve development flow in general.

6) Article tracking in arbitrary formats

This feature enables unexpected events to be recorded as "articles" along the time axis to aid in problem management. It also enables multiple items and formats to be defined for use in recording articles, which means that the information needed for tracking can be managed not just for problem management but also for risk management, test management, etc. Items and formats defined in this way can also be shared.

7) Article association

An article can be associated with other articles or with processes and deliverables so that the impact of a particular event can be quickly grasped.

8) Knowledge sharing and management within a project

As a development project progresses, various kinds of knowledge are created. Such knowledge is often generated at random times in no particular format. For this reason, we have prepared a Wiki-type site where project members can freely describe newly obtained knowledge.

9) “My Page” for displaying information for personal use

In the modern development process, work is becoming more specialized as technology becomes increasingly complex and sophisticated. Developers may thus find themselves working on several projects and carrying out different duties at the same time. To cope with such a situation, this function displays only those processes and articles assigned to a particular individual. This enables a developer to access needed information quickly and prevents him or her from overlooking measures that need to be performed.

5. Application examples

The P3 visualization system we developed has found widespread use within Fujitsu in the development of software and hardware products. We present here some application examples and their results.

5.1 Application to software development

In the software department, P3 is being applied to the development of middleware products running on hardware developed by Fujitsu. To facilitate this application, we established two working groups (WGs) called the

“development-process refinement WG” and “tool promotion WG” consisting of representatives from the various development departments. The development-process refinement WG reexamines existing development processes with an eye to exploiting the characteristics of IT-based tools, while the tool promotion WG provides tool-usage methods as well as templates for development processes refined by the development-process refinement WG. The combined activities of these two WGs have had tangible results: the number of standard development processes has been reduced by 35%, from 105 to 68, and the number of forms used in development work has been reduced by 53%, from 71 to 33. Furthermore, on measuring the effect of applying P3 to 37 products in fiscal year 2010 from a QCD perspective, a quality improvement of 35%, a cost reduction of 30%, and a delivery-period shortening of 27% on average was observed in those projects for which an effect could be observed (**Table 1**).

P3 was also applied to the development of ultra-high-speed in-memory data management software. This application exemplifies the visualization of the accumulation of knowledge and other know-how and the occurrence of problems within a project.

The developed software was used for the new trading system of the Tokyo Stock Exchange (TSE), which came on line in January 2010, to ensure high availability. When converting this middleware product into a general-purpose version, two major issues arose in the development process: first, how to shorten the development period given that the target was now

Table 1
Application effect (Software Business Group).

	Q (defect rate during testing)	C (total development cost)	D (development period)
Improved	13 products (35% quality improvement) Average reduction of 0.215 defects/KI	11 products (30% cost reduction) Average reduction of 5.5%	10 products (27% shorter delivery period) 4 months shorter on average
No change	21 products	20 products	23 products
Degraded	3 products Average increase of 0.023 defects/KI	6 products Average increase of 0.05%	4 products Average increase of 0.1 months

KI: Kilo step

a general, commercial product, and second, given that developers were being added or changed during the project, how to ensure development quality among developers and how to prevent obstacles to development that can be traced back from subsequent processes to previous processes (well-timed quality follow-up). To deal with these issues, project members were asked to use a P3 Wiki function to define development rules that clearly describe criteria for making progress in a process and for classifying bugs. This function was also used to record as know-how matters of interest discovered during the course of development. New project members could reference this information at any time. Project members could also perform a daily check of work progress registered using the P3 process management function and of problems registered using the P3 tracking management function so that no delays would occur in dealing with problems in development quality. As a result of these measures, development time was shortened by 75%, and the number of misunderstandings during development was reduced by 82% compared to the development project for the TSE's trading system version of this middleware. This had the effect of reducing the occurrence of problems: the number of reviews associated with redesign decreased by 45%, and the number of items pointed out by reviews decreased by 72%.

5.2 Application to hardware development

An example of applying P3 to hardware development is its application to the development of optical communications equipment in the network department. The recent spread of IT throughout society has increased the volume of traffic flowing through the network with the result that the function and performance requirements for new products have become quite severe. In the LSI department, the work of developing ASIC/FPGA products is centered about Fujitsu's Kawasaki plant and is shared among its Oyama plant and sites in Sendai,

Osaka, and Fukuoka. Several tens of products are developed every year in this way. Dealing with problems that occur after the shipping of hardware and especially ASIC products can incur considerable expense, more than tens of millions of yen at times. There is therefore a need to efficiently and effectively mitigate risk and resolve problems that occur during the course of development. Since sites sharing the work of the network department are distributed throughout the country, the conventional way of handling problems and difficulties was to have each site manage them using a spreadsheet program like Excel and forward the information so compiled to a coordinating department for analysis and checking. This method, however, made it difficult to obtain an overall picture of a certain problem in a prompt manner, so there was a need for an environment that could resolve this issue. The P3 tracking management function has made it possible for project members to determine in real time what problems have occurred at what time, which people are responding to those problems in what way, and what response is being made across all sites. Moreover, as development-related data is consolidated in one place, it has become easier to review past work after development has completed and to apply the information obtained to the next line of products.

6. Future developments

Over 6000 Fujitsu developers are now using the P3 system. This system enables all designers in a development project to manage problems and obstacles within the same environment, thereby promoting the sharing of a large amount of widely dispersed developer information and knowledge. It also enables the same environment and its method of use to be shared even among different projects, thereby minimizing the cost of sharing and managing information and knowledge.

The P3 system is now being enhanced so that it will work as illustrated in **Figure 3**. We are accumulating test items and test results

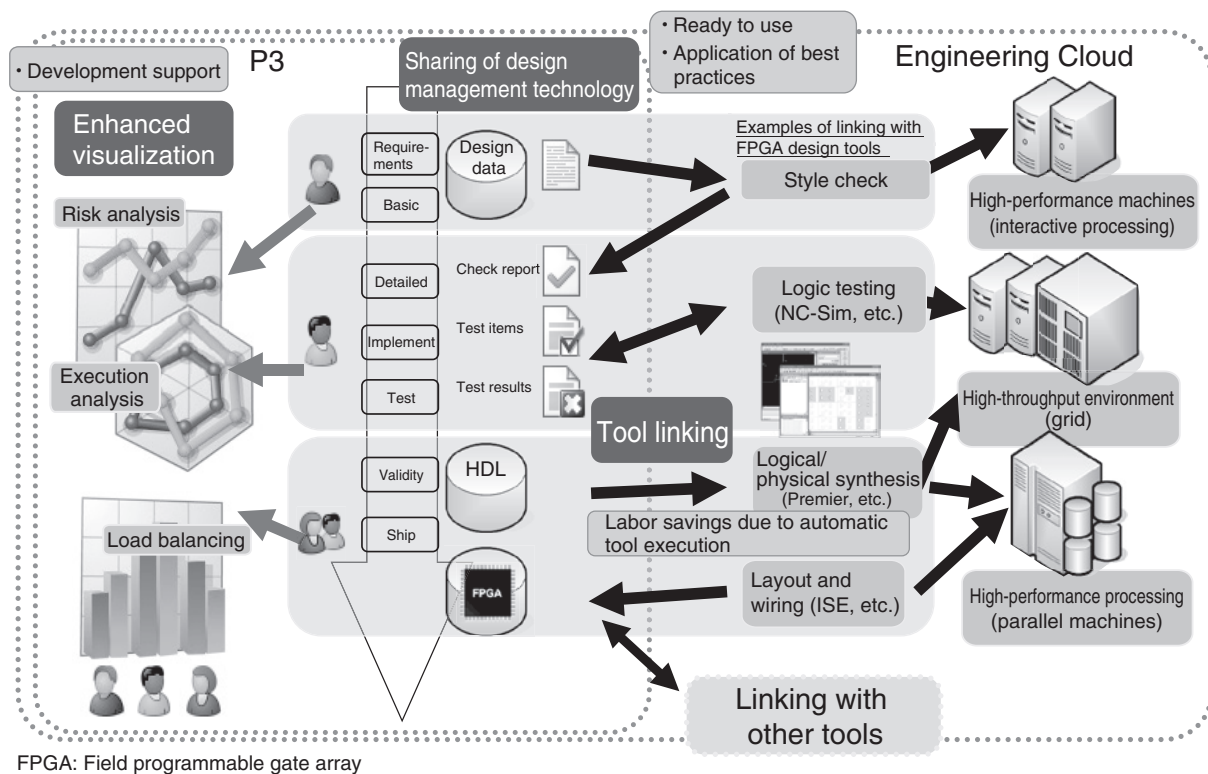


Figure 3
Enhancement of P3 system.

using the tracking management function and preparing a function for predicting the occurrence of problems in products under development and for giving advice on avoiding those problems. However, making accurate predictions requires the storing of a large volume of diverse types of information, and at present, this function is only being used for simple predictions that can be made on the basis of a relatively small amount of information. In a similar manner, it is also possible to perform risk management using the tracking management function, but we have not yet prepared a function for assessing the validity of risk extracted in this way. We plan to enhance such analysis functions and to develop a function for controlling a development environment based on virtual machines, which have recently been attracting attention. By making a seamless connection between process management and the development environment, it should be possible to automate a whole sequence of processes,

such as design rule checking at the backend, registration of those results with the tracking management function, and validity checking of the same results. This mechanism should have the effect of eliminating disparities in quality between individuals while raising efficiency in development work. We are also planning to create an environment for thoroughly applying best practices. This can be done, for example, by automatically recording when and how often developers reference development information/knowledge and by comparing the development conditions of a new project with those of similar, past projects. In this way, appropriate development knowledge and know-how can be provided as feedback to the new project.

7. Conclusion

This paper described how IT device development has become increasingly advanced and complicated, explained the problems involved

in sharing the information and knowledge generated in the development process, described the requirements for solving those problems, and introduced the Project Propulsion Platform (P3) visualization system developed on the basis of those requirements.

We showed that using this platform in software and hardware development can effectively improve development efficiency and productivity. Looking forward, we plan to make further improvements to this platform with an eye to achieving more innovations in the development process.

References

- 1) International Business Strategies, Inc.
<http://www.ibs-inc.net/>
- 2) H. Kurosaka et al.: System Level Design Technique based on Analysis of Research and Standardization Trends. (in Japanese), Proceedings of the DA Symposium 2003, Information Processing Society of Japan, pp. 183–196, 2003.
- 3) M. Imai: Current Status and Future Trends of Multiprocessor SoC. (in Japanese), 18th Shin-Yokohama IT Cluster Exchange Meeting, 2007.
- 4) T. Furuyama et al.: Analysis of the Factors that Affect Performance of Software Projects. (in Japanese), IPSJ SIG Technical Reports, Software Engineering 2006 (75), pp. 47–54 (2006).
- 5) Y. Yamamoto: Device Survey (11)—Has the difficulty of FPGA design exceeded that of ASIC design? (in Japanese).
<http://www.kumikomi.net/archives/2010/04/co07fpga.php>
- 6) M. Yamada: Software as Executable Knowledge, “metabolics,” (in Japanese).
<http://www.metabolics.co.jp/>
- 7) W. Royce: Managing the development of large software systems. Proceedings IEEE WESCON, pp. 1–9 (August 1970).
- 8) B. Boehm: A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, Vol. 11, pp. 22–41 (1988).
- 9) International Organization for Standardization (ISO): ISO9001-2008 Quality management systems—Requirements, ISO9001.
- 10) Software Engineering Institute: CMMI.
<http://www.sei.cmu.edu/cmmi/index.cfm>
- 11) PMI Japan: PMBOK Guide, Fourth Edition Reference, 2011, (in Japanese).



Tomonori Yamashita
Fujitsu Advanced Technologies Ltd.
Mr. Yamashita is engaged in the development of project management systems.



Masaru Kameda
Fujitsu Ltd.
Mr. Kameda is engaged in the development of ASIC/FPGA devices for optical communications equipment.



Kimio Saito
Fujitsu Ltd.
Mr. Saito is engaged in raising the efficiency of middleware-product development.