Promoting Global Software Factory Activities through "Yakushin," Integrated Project Infrastructure

● Naokazu Hazeyama ● Hideki Sato ● Hiroshi Tahata ● Masaki Yamashita

For businesses in government and public enterprise sectors, it is necessary to build high-quality systems quickly and at low cost that correspond to specific laws, institutions, and the operating characteristics and required specifications for each business. In addition, the approach to system development has changed from vendor-provided development solutions to ones that combine vendor-provided and open source software solutions. On the basis of these circumstances, when Fujitsu is developing and operating public-related systems, it promotes activities focused on determining development approaches and deciding project management approaches, with a priority on stable operation of customer systems. Starting in FY2017, we launched global software factory activities (software industrialization from a global perspective) and the internal application "Yakushin," an integrated project infrastructure to standardize work quality. Yakushin enables us to determine development approaches and tools, as well as project management approaches. In this paper, we describe Yakushin's core development infrastructure, Yakushin/Crust, as well as its project integration management infrastructure, Yakushin/Tophat.

1. Introduction

System development for businesses in government and public enterprise sectors differs in regard to methods and development time due to specific laws, frameworks, and characteristics and requirements/specifications of the target businesses. As a result, it is not easy to apply solution-oriented development using existing business-system know-how and business packages, and there are many cases of individually commissioned made-to-order developments in this field.

In recent years, it has become necessary to adapt to changing social circumstances and legal reforms quickly, which is making system development increasingly complicated and delivery times shorter. There is also a need for high-quality systems incorporating advanced security functions and setting up business rules for linking ministry and agency systems.

In Japan, Cabinet approval of the "Declaration to Be the World's Most Advanced Nation and Basic Act on the Advancement of Utilizing Public and Private Sector Data"[1] in May 2017 is driving the conversion of various types of data held by public and private enterprises to an open-data format. To use such data, there is a need for cross-sector search functions, and studies are now being conducted on the development of public-private systems of engagement (SoE). To this end, there is a need for even swifter system delivery by performing proof of concept (PoC) and proof of business (PoB) demonstrations in a short period of time and evaluating their results by repeated execution of the plan-do-check-act (PDCA) cycle.

It is also necessary to adopt a mechanism for reusing various types of system development methods and knowledge gained from completed projects as practical wisdom. For example, if good case studies and useful knowledge obtained during a development project could be swiftly applied to other projects through horizontal development and sharing (integration), it would be possible to create high-quality systems for customers with short delivery times on an ongoing basis.

To put such ideas into practice, we have constructed a business model for remote development, operation, and support of government-related customer systems under our care. Furthermore, in FY2017, we initiated global software factory activities (software industrialization from a global perspective) and launched the internal application of "Yakushin," an integrated project infrastructure, to standardize work quality. The name "Yakushin" originates from the Japanese word for "rapid progress" and signifies dramatic evolution together with customers while applying existing system integration (SI) activities.

In this paper, we describe in detail in-house activities at Fujitsu for putting Yakushin into practice.

2. Activities to standardize work quality

Open source software (OSS) has a broad range of use in application development, and in addition to previously developed technology, development techniques using new technologies are necessary. It is also necessary that vendors use such technologies to continuously provide highly-productive, high-quality measures for manufacturing. At the same time, the operating environment of business systems is diversifying—today they include on-premises, private cloud, and public cloud services, or a combination of them called hybrid clouds. Vendors must therefore adopt an evolutionary and extendible development style that can meet customer requirements while achieving portability between any operating environments.

For project management, communication and management methods are becoming major issues in operations as the scale of development escalates. As a result, there is a need for progress management that can synchronize project statuses and actual conditions on the basis of reports and deliverables, respectively. For resources, the use of external ones may occur, as in offshore outsourcing in which development work is consigned to an overseas firm, or near-shore outsourcing in which some or all work is consigned to a firm that is relatively close in terms of distance. In either case, project organization is such that work is not necessarily performed at the same place, so a mechanism is needed to bridge the communication gap. Project organization is also needed to detect problems as soon as possible and help project managers focus their efforts on solving those problems.

In response to these needs, Fujitsu has been constructing an infrastructure that controls the manner of development and project management from the head office while excluding individual skills and attributes. Repeated use of this infrastructure (reuse of practical wisdom) will make it possible to continually provide customers with high-quality, highly-productive systems. Specifically, we began development of an integrated project infrastructure for public systems in 2014 and renamed those activities "Yakushin" (**Figure 1**) with

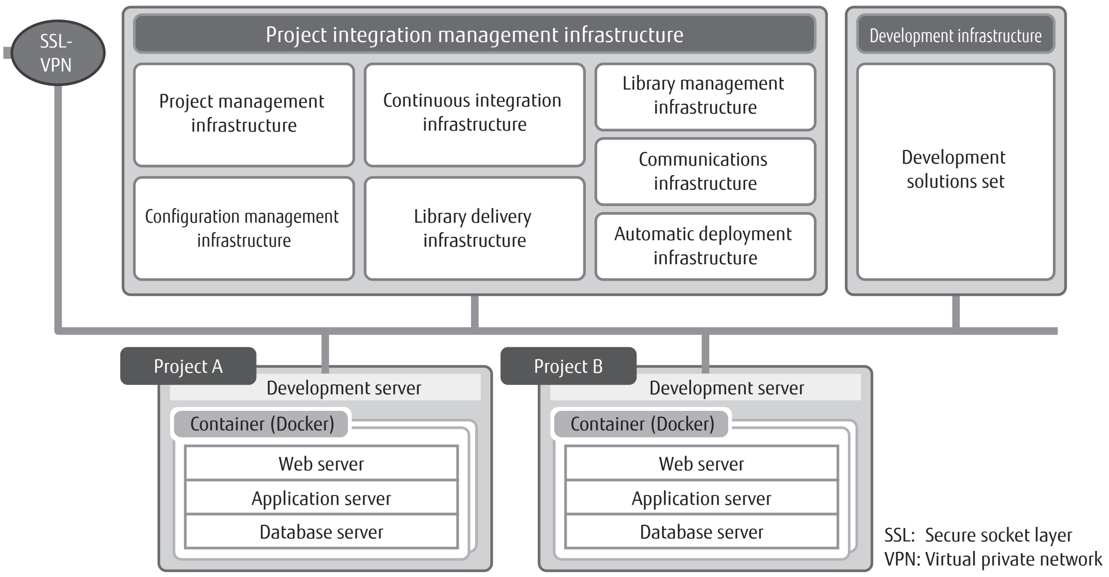

SSL-VPN
Project integration management infrastructure
Development infrastructure
Project management infrastructure
Continuous integration infrastructure
Library management infrastructure
Communications infrastructure
Configuration management infrastructure
Library delivery infrastructure
Automatic deployment infrastructure
Development solutions set
Project A
Development server
Container (Docker)
Web server
Application server
Database server
Project B
Development server
Container (Docker)
Web server
Application server
Database server
SSL: Secure socket layer
VPN: Virtual private network

Figure 1
Overall view of Yakushin.

the launch of the Global Software Factory (GSF) office in April 2017.

The GSF office was launched for the purpose of passing on the SI development know-how of aging systems engineers (SEs), migrating to a cost-saving scheme by making extensive use of global personnel, and reforming the SE work style.

At the GSF office, the following infrastructures must be constructed to continue Fujitsu's SI business into the future:

- Development infrastructure
- Project integration management infrastructure

In this paper, we describe in detail Yakushin's two main components for achieving the above: Yakushin/Crust, a development infrastructure at the core of Yakushin, and Yakushin/Tophat, a project integration management infrastructure.

3. Yakushin/Crust: development infrastructure

"Container virtualization" has become increasingly popular in recent years as a technology for controlling and managing business applications and their operating environments. Situated at the OS-level of a server, container virtualization makes it possible to allocate server resources to each business-application operating environment and efficiently package and operate a variety of applications. The way in which a business application itself is created depends not only on the operating environment but also on frameworks and business rules, and as a result, the environment information needed for application operation impedes the packing necessary for container virtualization.

Yakushin/Crust is a development infrastructure that enables the packing of business applications. Short for "containerization runaway solution of application development technology," Crust provides specific measures on the application side for container virtualization that is expected to grow in popularity from here on. It also includes many development solutions that can be easily and quickly deployed. The following describes specific containerization measures for business applications in Crust.

1) Loose coupling of business applications

The mechanism of a business application consists of seven functions that operate using a set of transaction data: search conditions, search results, register, update, browse, delete, and comma separated value (CSV) download. Functions customized to an actual business system may also be included. The basic processing pattern of Crust is shown in **Figure 2**. Crust includes template source code that implements

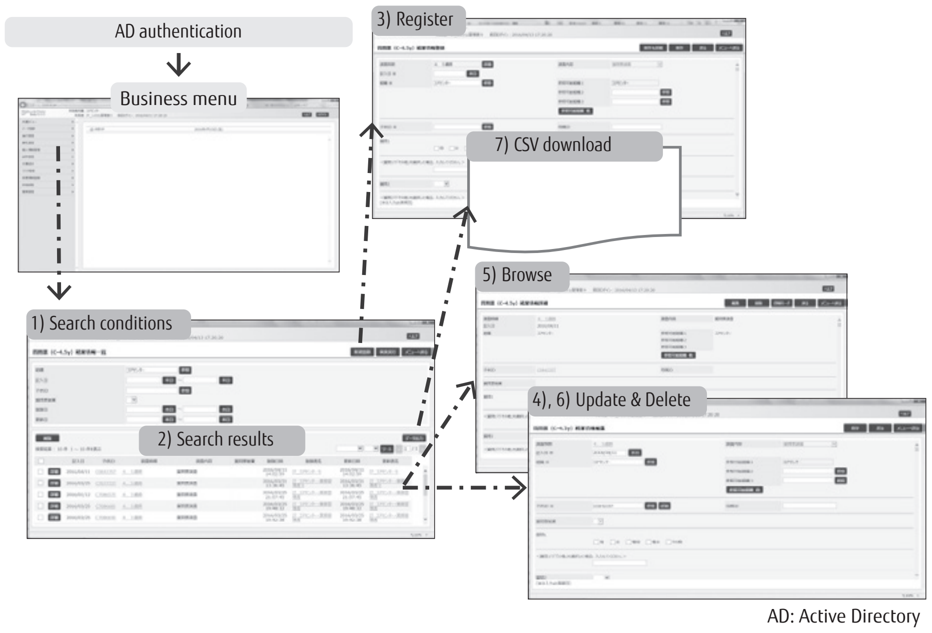


Figure 2
Basic processing pattern.

this basic processing pattern. In addition to application control within the OSS framework, the template includes a standardized business logic section. A developer need only fill in a design sheet to embody the template into the program code and generate a business application by using the Crust application builder.

This template was developed based on Spring, an open source application framework for Java platforms, but other frameworks are also supported. To replace Spring with another framework, it is only necessary to prepare a template corresponding to that framework and re-execute the application builder. This method is meant to facilitate reconstruction work in future system upgrades.

2) Loose coupling of business logic layers

A business application takes form by reflecting the customer's business in system specifications and translating those specifications into program code. Business model specifications can be broadly divided into those that change with the times and those that do not. In this regard, describing processes or values in specifications that can easily be changed directly in the source code (hard coding) adds to the time needed to make revisions, thereby, hindering the customer's business operations. This problem is resolved by using a business rule management system (BRMS) (rule engine) that externally defines rules found within business model specifications separate from business logic (**Figure 3**).

Such externally defined rules constitute a document as a format that can be described by even non-programmers. Crust provides as standard a mechanism for inserting and driving externally defined rules in parts of the program that code business logic.

3) Loose coupling of runtime environment layers

There are many definitions dependent on the execution environment in the case of business applications. The following lists some of these.

- Connection information with other systems (test/actual use)
- Inter-system file information (paths, definitions)
- Language notation information (Japanese/English)
- Batch execution information (execution time, execution conditions)
- Log collection information (log collection level, collection information)

The packing process in container virtualization proceeds while locations dependent on each runtime environment in the upstream processes are designed. In addition, compiling design information in a design sheet provided by Crust creates a containerized state for each runtime environment simultaneously with the application build-and-deploy process (**Figure 4**). This mechanism helps to decrease project work time and prevent errors at the time of application building through environment building automation. CD and CI in the project integration management infrastructure described below is one example of such automation.

4) Loose coupling of external public service layers

We have taken on system development contracts from public entities and their auxiliary organizations

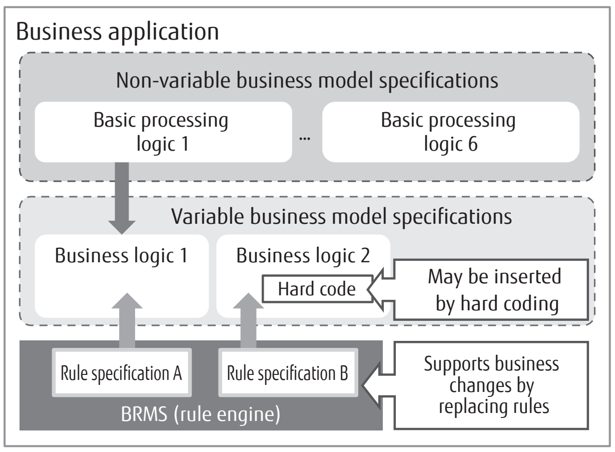


Figure 3
Using BRMS to achieve functions.

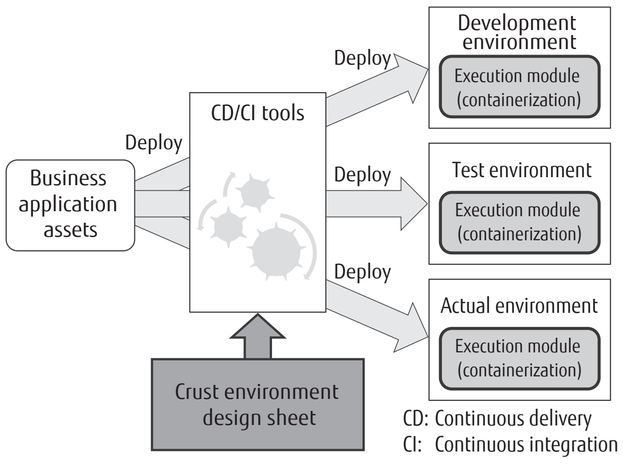
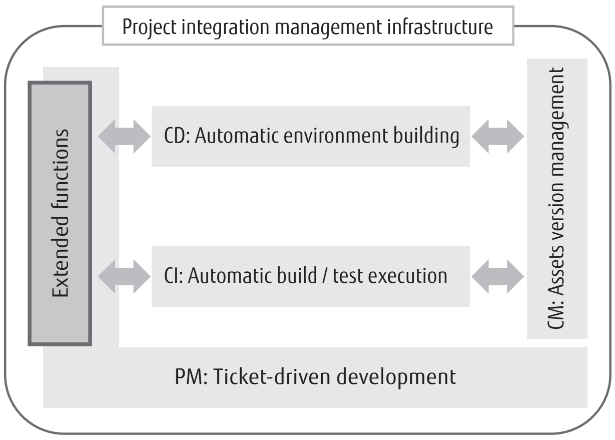

Development environment
Execution module (containerization)
Deploy
CD/CI tools
Deploy
Business application assets
Deploy
Test environment
Execution module (containerization)
Deploy
Actual environment
Execution module (containerization)
Crust environment design sheet
CD: Continuous delivery
CI: Continuous integration

Figure 4
Container deployment in each runtime environment.

in Japan. In system development projects to date, we have found that a number of common programs exist among the system requirements of various ministries and agencies. Crust prepares such common programs as service components that can be used across multiple systems and provides them in a usage format independent of platform.

For example, linking public and private financial systems can be difficult when attempting to establish business rules related to a system of collection and benefits payments. In such a case, highly reusable functions can be packed, and service components can be set up as web application programming interfaces (APIs) to eliminate the need of implementing them separately for each project. This approach helps to maintain the quality of customers' mission-critical business systems.


Project integration management infrastructure
Extended functions
CD: Automatic environment building
CI: Automatic build / test execution
CM: Assets version management
PM: Ticket-driven development

CM: Configuration management
PM: Project management

Figure 5
Yakushin/Tophat configuration.

4. Yakushin/Tophat: project integration management infrastructure

To achieve a project life cycle as required by the customer, attention must be focused on the manufacturing process in the operations of a development project. Yakushin/Tophat is a project integration management infrastructure that suppresses individual skills and attributes in project management and achieves real-time management of an entire project through the visualization of work status. "Tophat" is short for "Ticket-driven operation for promoting high-able total management project."

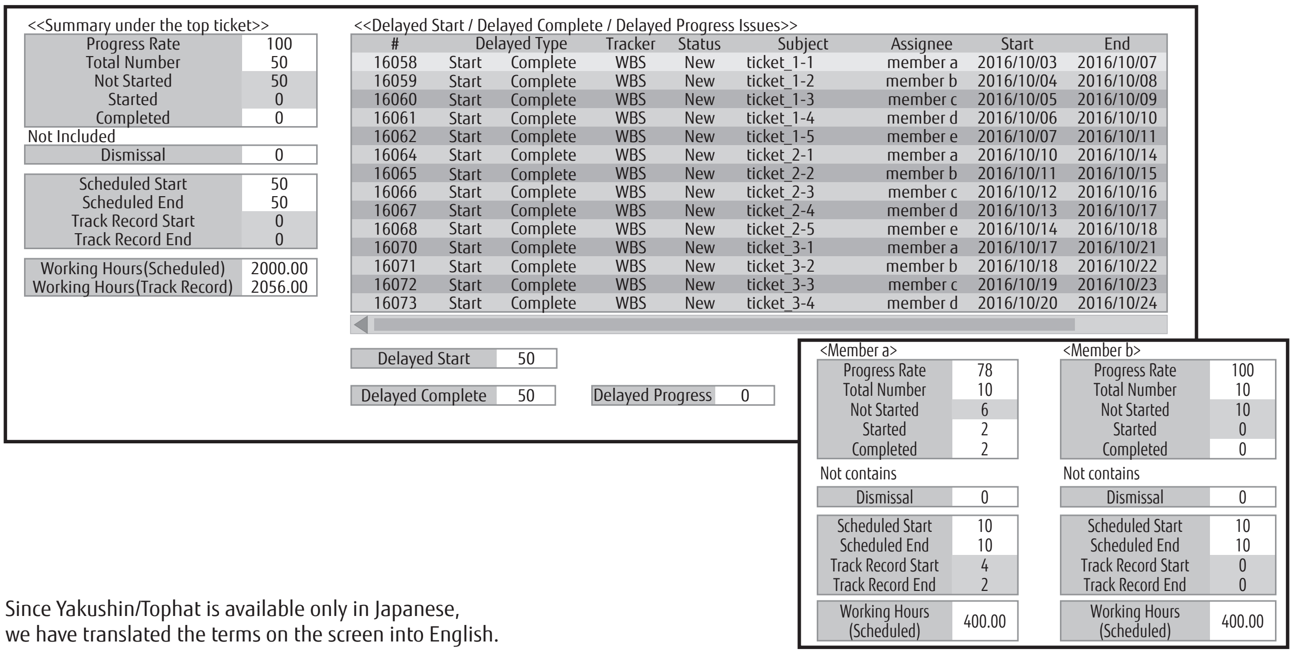
With Tophat, we have been able to absorb the information necessary for management from activities at development sites and visualize that information with a view to promote remote distributed development.[2] We have also set up a "ticket-driven management infrastructure" (**Figure 5**) as a project management system, having a mechanism for discovering problems early. The use of tickets containing information on the work breakdown structure (WBS), deliverables, test items, issues, etc., can stimulate communication among project members. In short, it can achieve global project control that mitigates location, time, and language constraints.

The following describes the functions provided by Yakushin/Tophat.

1) Real-time, ticket-driven work management independent of reports from people in charge
- Registers work items of each supervisor in the system and performs detailed management of estimated and actual results by linking with ticket information
- Determines work status at any time using a function for tabulating WBS task statuses (**Figure 6**) and an earned value management (EVM) function in conjunction with ticket statuses
- Determines test progress at specific times using a test management function and quality status using a curve representing the decrease of remaining programs to be checked and the increase of detected bugs (P/B curve) and fault-status tabulation

2) Determination of overall quality status
- Integrates and registers quality control criteria such as record-of-review slips and problem slips into the system
- Incorporates a quality analysis function into the system and determines current quality status even during a process in progress
- Determines process-specific change in quality by integrating quality control criteria
- Refines quality indices and performs statistics-based quality benchmark evaluations by repeatedly collecting results by the same management method

3) Integrated management of development assets with quality improvement
- Systematically manages original application



<<Summary under the top ticket>>

Progress Rate	100
Total Number	50
Not Started	50
Started	0
Completed	0

Not Included

Dismissal	0

Scheduled Start	50
Scheduled End	50
Track Record Start	0
Track Record End	0

Working Hours(Scheduled)	2000.00
Working Hours(Track Record)	2056.00

<<Delayed Start / Delayed Complete / Delayed Progress Issues>>

#	Delayed Type		Tracker	Status	Subject	Assignee	Start	End
16058	Start	Complete	WBS	New	ticket_1-1	member a	2016/10/03	2016/10/07
16059	Start	Complete	WBS	New	ticket_1-2	member b	2016/10/04	2016/10/08
16060	Start	Complete	WBS	New	ticket_1-3	member c	2016/10/05	2016/10/09
16061	Start	Complete	WBS	New	ticket_1-4	member d	2016/10/06	2016/10/10
16062	Start	Complete	WBS	New	ticket_1-5	member e	2016/10/07	2016/10/11
16064	Start	Complete	WBS	New	ticket_2-1	member a	2016/10/10	2016/10/14
16065	Start	Complete	WBS	New	ticket_2-2	member b	2016/10/11	2016/10/15
16066	Start	Complete	WBS	New	ticket_2-3	member c	2016/10/12	2016/10/16
16067	Start	Complete	WBS	New	ticket_2-4	member d	2016/10/13	2016/10/17
16068	Start	Complete	WBS	New	ticket_2-5	member e	2016/10/14	2016/10/18
16070	Start	Complete	WBS	New	ticket_3-1	member a	2016/10/17	2016/10/21
16071	Start	Complete	WBS	New	ticket_3-2	member b	2016/10/18	2016/10/22
16072	Start	Complete	WBS	New	ticket_3-3	member c	2016/10/19	2016/10/23
16073	Start	Complete	WBS	New	ticket_3-4	member d	2016/10/20	2016/10/24

Delayed Start	50
Delayed Complete	50
Delayed Progress	0

<Member a>		<Member b>	
Progress Rate	78	Progress Rate	100
Total Number	10	Total Number	10
Not Started	6	Not Started	10
Started	2	Started	0
Completed	2	Completed	0
Not contains		Not contains	
Dismissal	0	Dismissal	0
Scheduled Start	10	Scheduled Start	10
Scheduled End	10	Scheduled End	10
Track Record Start	4	Track Record Start	0
Track Record End	2	Track Record End	0
Working Hours (Scheduled)	400.00	Working Hours (Scheduled)	400.00

Since Yakushin/Tophat is available only in Japanese, we have translated the terms on the screen into English.

Figure 6
Task tabulation function.

assets and visualizes change history to provide asset status at any time
- Incorporates a CI mechanism; automatically compiles, deploys, and tests when registering, changing, and deleting assets; and detects inconsistencies or deterioration early
- Determines on-site progress and quality status by linking number of pages of registered assets with scale measurements and program static analysis
- Achieves seamless deliverable testing with the testing department

4) Control by standard management practices
- Shares practices that describe management procedures and workflows with project participants
- Supports early project launching through the operating procedures of ticket-driven management tools provided by Yakushin

5) Control based on work guidelines
- Presents work guidelines describing design documentation templates, process completion criteria, etc., and unifies management and development processes

Yakushin provides these project standards as a "project integration development infrastructure" on the level of an SE knowledge base. In addition to a ticket-driven management infrastructure, Yakushin consolidates the Yakushin/Crust infrastructure described above, groups of common business components, and rental functions for development environments and operating environments, such as PoC demonstrations in a development environment portal. All of the above are provided in a uniform manner to developers. Rental functions for development/operating environments make use of container virtualization technology on Fujitsu's private cloud for development. Software component groups such as middleware, frameworks, and management tools that are needed to complete testing are packaged and made available for immediate deployment.

In addition, FUJITSU Cloud Service K5, Fujitsu's public cloud service that lies at the core of FUJITSU Digital Business Platform MetaArc, may be selected for the operating environment. MetaArc is a development and execution environment for early deployment of digital business, and preparations proceed using K5 and MetaArc to enable seamless linking with various technologies such as AI and IoT that constitute stored knowledge.[3)]

5. Conclusion

We described the activities of the Yakushin integrated project infrastructure that is being internally applied at Fujitsu. Our plan is to continue these activities to provide high-quality ICT systems and services that our customers and society can rely on. Going forward, we will strive to improve our technical competence and make full use of the business know-how and experiences that we have accumulated to date as an SE group that is always up for a challenge.

Finally, we would like to extend our deep appreciation to all concerned at KEIKEN SYSTEM Co., Ltd. for their gracious cooperation in global software factory activities.

References

1) Cabinet Secretariat: Declaration to Be the World's Most Advanced Nation and Basic Act on the Advancement of Utilizing Public and Private Sector Data (in Japanese). *http://www.itdashboard.go.jp/Achievement*
2) J. Watanabe et al.: Software Development Industrialization by Process-Centered Style. FUJITSU Sci. Tech. J., Vol. 46, No. 2, pp. 168–176 (2010). *http://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol46-2/paper05.pdf*
3) N. Nakamura: Fujitsu's Leading Platform for Digital Business. FUJITSU Sci. Tech. J., Vol. 53, No. 1, pp. 3–10 (2017). *http://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol53-1/paper01.pdf*

Naokazu Hazeyama
Fujitsu Ltd.
Mr. Hazeyama is currently engaged in development and management of large-scale public SI projects.

Hideki Sato
Fujitsu Ltd.
Mr. Sato is currently engaged in system development for the public sector.

Hiroshi Tahata
Fujitsu Ltd.
Mr. Tahata is currently engaged in management of small-scale to large-scale development projects in the public market.

Masaki Yamashita
Fujitsu Applications, Ltd.
Mr. Yamashita is currently engaged in organization and application of development technology in software development.