Standardization of Development Process and Additional Efforts Focusing on Web Application Development

● Ryoko Saito ● Satoru Okiyama ● Fusami Hirai

(Manuscript received March 14, 2006)

In this time of remarkable informationization, information systems have become an essential business infrastructure for managing corporate activities. Under these circumstances, due to the highly diverse functions and the related highly divisionized and specialized work, a large number of field workers are involved in system development. They must therefore communicate properly and cooperate in work through a common understanding. They must also share their roles systematically and clarify the range of responsibilities between themselves. To help meet this need, Fujitsu has developed the Solution-oriented system Development Engineering Methodology 21 (SDEM21), a system development map that plainly systematizes the tasks of an entire lifecycle, from planning to operation and maintenance. Web application development has accelerated in recent years, and to assist in this area, Fujitsu has developed ComponentAA Development Method. This method makes it easy to construct a flexible project development standard for various technologies for implementing Web applications. Moreover, Fujitsu has developed Secure Web Application, which is a security countermeasures standard for building stable and secure Web services. This paper introduces SDEM21, ComponentAA Development Method, and Secure Web Application.

1. Introduction

In this time of remarkable informationization, information systems have become an essential business infrastructure for managing enterprise activities. For this reason, many personnel in top management and user departments are involved in their development. At the same time, the importance of security countermeasures within these systems is growing.

This paper introduces Solution-oriented system Development Engineering Methodology 21 (SDEM21) to explain the standard procedures of system development in Fujitsu. It also introduces ComponentAA Development Method and Secure Web Application and includes some viewpoints of Web application development, which is currently the predominant system type.

2. SDEM21

This section gives an overview of SDEM21 and some materials that enable SDEM21 to be used more practically.

2.1 Overview of SDEM21

Due to the highly diverse functions and the related highly divisionized and specialized work, workers must cover various fields. To succeed in system planning, development, operation, and maintenance, workers must communicate properly and cooperate in work through a common understanding. They must also share their roles systematically and clarify the range of responsibilities between themselves.

SDEM21 plainly systematizes the tasks of an entire lifecycle, from planning to operation and maintenance. In other words, it visualizes system development. It was developed based on the WG activities of on-site system engineers and the numerous project experiences of Fujitsu and complies with the standards of the International Standards Organization (ISO).

The SDEM21 matrix and the phases and categories of SDEM21 are described below.

2.1.1 Outline of SDEM21 matrix

Phases such as design, implementation, and testing are placed on the horizontal axis, while categories such as business, infrastructure, operation and migration, which classify the tasks for development, are placed on the vertical axis. The tasks to be performed (defined in the work breakdown structure [WBS]) are defined at the points where the axes meet (**Figure 1**).

By using this matrix and plainly showing the entire view of the diverse tasks to all workers involved, full coverage can be ensured.

2.1.2 Phase and V-model for quality assurance

Phases are defined to manage the process in a continuous sequence of tasks in system planning, development, operation, and maintenance (**Table 1**). Phases are set based on the V-model for quality assurance (hereafter called the V-model).

The V-model associates the integration process with the breakdown process to assure quality (**Figure 2**). By pairing the testing phase and design phase based on system components, it makes verification and/or validation items in the testing phase that correspond to the design phase more specific and therefore enables quality to be

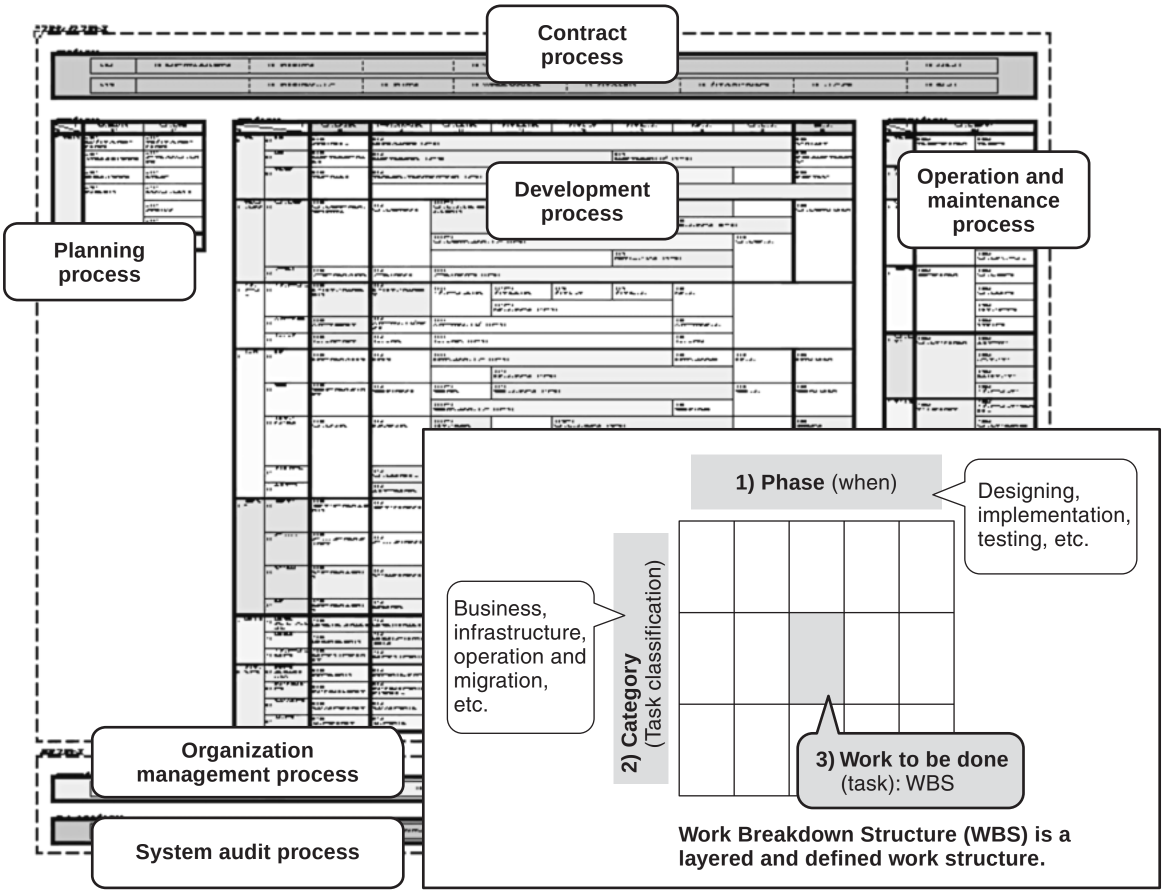

Contract process
Development process
Operation and maintenance process
Planning process
Organization management process
System audit process
1) Phase (when)
Designing, implementation, testing, etc.
2) Category (Task classification)
Business, infrastructure, operation and migration, etc.
3) Work to be done (task): WBS
Work Breakdown Structure (WBS) is a layered and defined work structure.

Figure 1
SDEM Matrix.

Table 1
Definition of phases.

Planning		Information vision planning	VP	• Develop information vision strategy, determine execution priorities, and develop mid-term or long-term plans.
		System planning	SP	• Analyze current business system, define requirements, evaluate investment effects, and make decision on development.
Development	Design	System architectural design	SA	• Check requirements for systemization, and decide range of systemization through business analysis. • Design system architecture and study feasibility. • Develop project plan, and prepare method for managing project.
		User interface design	UI	• Design business system specifications (process function, data structure, screens, forms). • Develop detailed system architectural design, and design means or operation and migration. • Develop overall test plan.
		System structure design	SS	• Break down process into programs, determine internal structure of system, and design common programs. • Design operation management system, security system, and migration tool. • Develop system test plan and operational test plan.
	Implementation	Program structure design	PS	• Determine program structure, and define logic.
		Programming	PG	• Develop program based on program structure design, and check the operation.
		Program test	PT	• Conduct test based on program test specifications, and verify quality.
	Testing	Integration test	IT	• Integrate programs, perform tests for each process, and verify quality. • Conduct interface tests between all processes, including interface for external systems. • If necessary, divide IT phase based on integration level or processing type (e.g., online or batch).
		System test	ST	• Test business system functions on actual machine. • Conduct overall system verification for performance, reliability, operability, security, etc.
		Operational test	OT	• Conduct trial operation by user departments using machine used in operation, environment, and data. • Validate business system functions, performance, reliability, operability, security, etc. • Make decision on migration to start of operations and migrate business.
Operation and maintenance		System operation and maintenance	OM	• Check means for business operation management, user department support, system operation, operation management, and system maintenance. • Deal with problems and Q&As, manage modifications of network, hardware, software, application specifications, etc. and develop improvement plan.

ensured. Needless to say, it is essential to perform verification and/or validation using reviews and/or prototyping at each design phase and to perform quality evaluation and judgment at each testing phase.

2.1.3 Categories

A category is a large classification of the tasks required to execute each process. Categories are classified based on the relationships between tasks and the knowledge and technology required to perform tasks. They can be used to determine the roles of the workers involved in planning, development, operation, and maintenance and to build teams in the project.

For example, the categories in the development process are business, business system specification, application, infrastructure, operation and migration (including security), development support, and project management.

By classifying the tasks as described above, omissions in task management, task estimation, and role assignment are prevented, while progress management is also enabled. In addition, by estimating the volume of tasks, setting quality targets, and managing the results based on specific task classifications, it becomes easier to measure the extent of progress. Furthermore, systematized tasks can be used as a basis for human resource development (e.g., when training

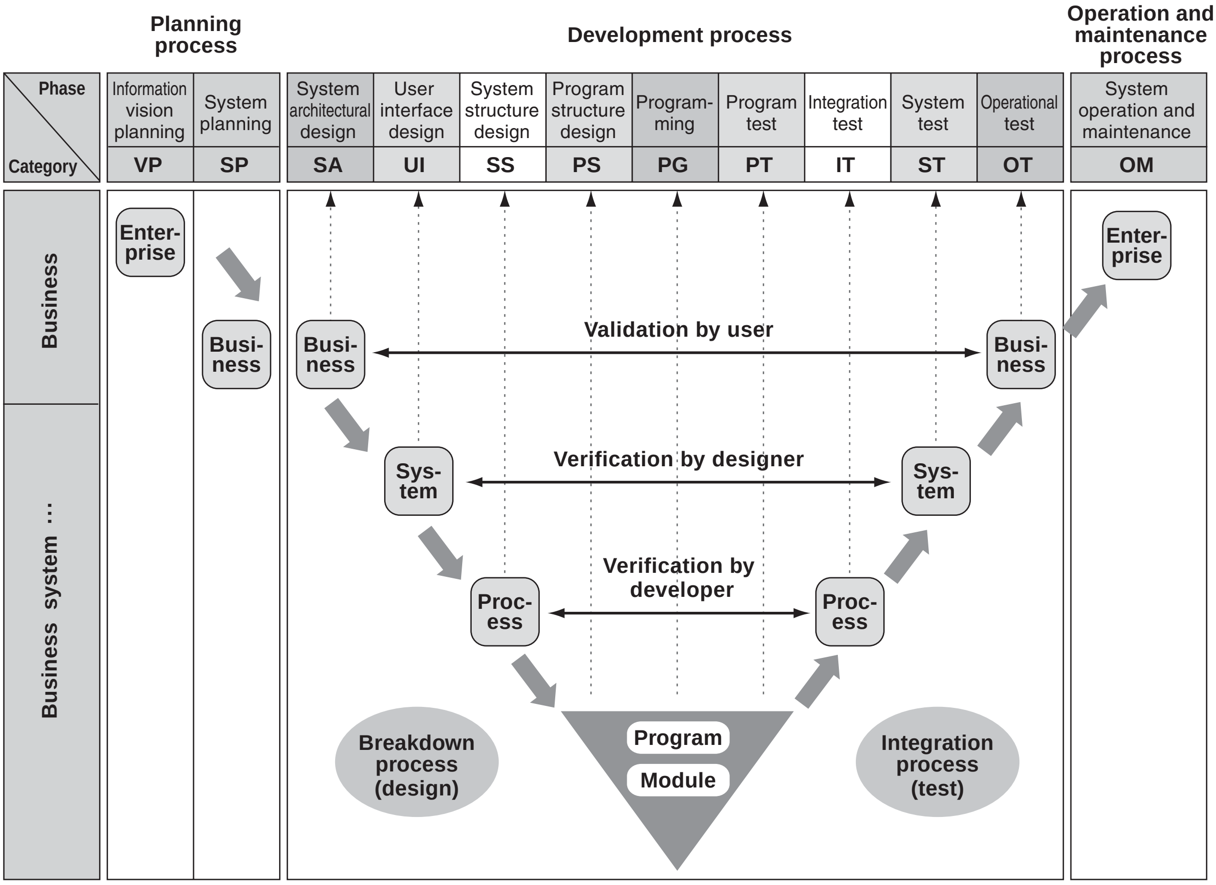


Figure 2
V-model of quality assurance.

new and junior employees or defining the image of an ideal employee) and improving processes (as a basis for evaluation when seeking improvements, etc.).

2.2 Materials for using SDEM more practically

The time allowed to develop a system in a project is becoming shorter, and there is little time to spend on initiating a project. To address this situation, codes of conduct that define the required actions and practical development standards that a project can use with the least possible workload are prepared for system engineers.

2.2.1 SDEM practical handbook (codes of conduct)

This handbook summarizes three principles that describe 1) how to cope with tasks, 2) the actions to be taken based on the results of tasks, and 3) the know-how in development tasks for each theme required to execute projects. The handbook covers 16 themes, based on SDEM matrix, for example, project planning, application architecture, and system architecture.

2.2.2 SDEM practical development standards

These are practical development standards that specify task sequences proceduralized from WBS using PERT, practical WBS descriptions, and document templates. They are prepared for three areas: 1) general business application development (non-Web application, COBOL/C, conventional notation), 2) ComponentAA Development Method (Web application, Java, Unified

Modeling Language [UML] notation) for developing business applications, and 3) an infrastructure task kit for developing a system infrastructure.

3. ComponentAA Development Method

ComponentAA Development Method[1)] is intended to provide a consistent development standard covering various technologies of Web applications. It consists of standards for development task procedures, document standards, and various types of development techniques (**Figure 3**). The following section gives a brief explanation of how ComponentAA Development Method supports Web application technologies.

3.1 Dealing with implementation technologies of Web applications

It is difficult to apply development standards of mainframe and client/server systems to the development of Web applications. This is because the application structure differs depending on the assumed implementation technologies (e.g., middleware and frameworks), so the design items and objects to be implemented are completely different from those for Web application development. If these development standards are applied forcibly, a substantial change in the development standards is inevitable. In addition, there are several technology options for implementing Web applications. For example, Servlet/JSP technology or Applet technology can be used for implementing a screen. For this reason, a change of design items occurs depending on the technology selected. This is another reason why the construction of development standards for Web applications is a difficult task.

ComponentAA Development Method directs developers to build a system by assembling components so that the structure of those components is arranged in the following three layers:[2)] the presentation layer, which displays and controls screens; the control layer, which executes application logic; and the model layer, which is responsible for accessing databases and

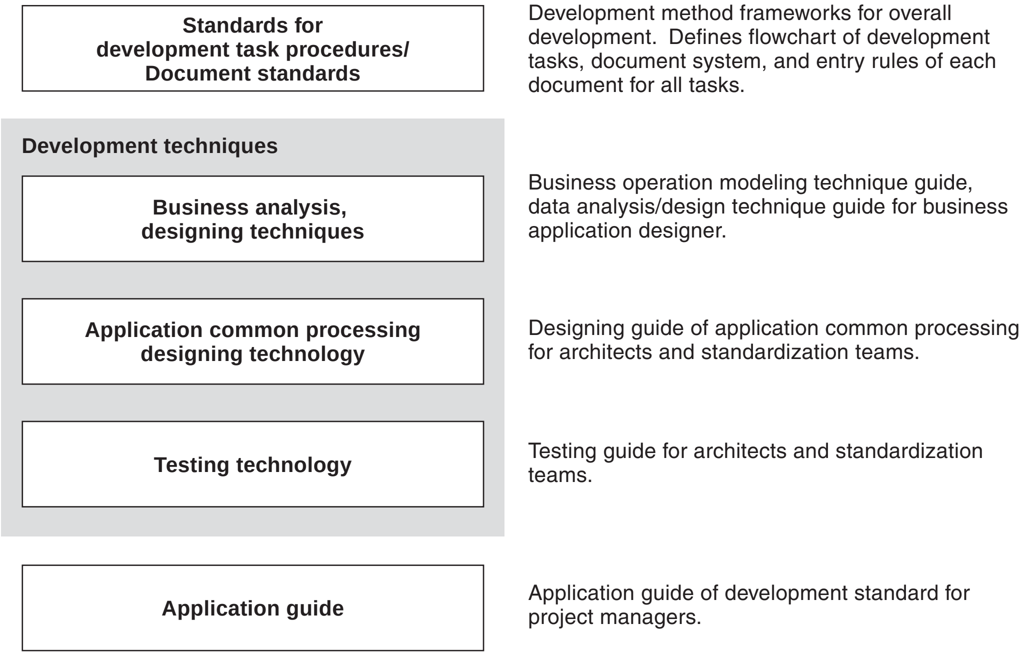

Standards for development task procedures/ Document standards
Development method frameworks for overall development. Defines flowchart of development tasks, document system, and entry rules of each document for all tasks.
Development techniques
Business analysis, designing techniques
Business operation modeling technique guide, data analysis/design technique guide for business application designer.
Application common processing designing technology
Designing guide of application common processing for architects and standardization teams.
Testing technology
Testing guide for architects and standardization teams.
Application guide
Application guide of development standard for project managers.

Figure 3
Technology system of ComponentAA Development Method.

maintaining data consistency. This structure is appropriate for implementing Web applications. To facilitate the development of Web applications, ComponentAA Development Method defines design procedures and design documents for the components of each layer (**Figure 4**). It also provides design guides for each selectable technology. For example, when Servlet/JSP technology is applied to the presentation layer, design guides for Servlet/JSP are used, and design guides for Applet can be used when applying Applet technology. By combining these rules and design guides, it becomes possible to deal with various technologies for implementing Web applications and easily constructing development standards suited for the project.

3.2 Using UML

UML[3] is a widely used general notation for object-oriented analysis and design. However, when applying UML, it cannot be used effectively if the purpose and method of use are not clearly defined in the specific development procedures. For example, if designers are simply asked to design classes using UML class diagrams, the contents and quality of the completed design documents will vary and development will not be efficient.

ComponentAA Development Method provides a specific purpose and method for using UML. For example, it states that a screen designer should write a UML activity diagram in the user interface design phase in order to design a screen transition diagram (i.e., it defines who writes what and also defines when and why

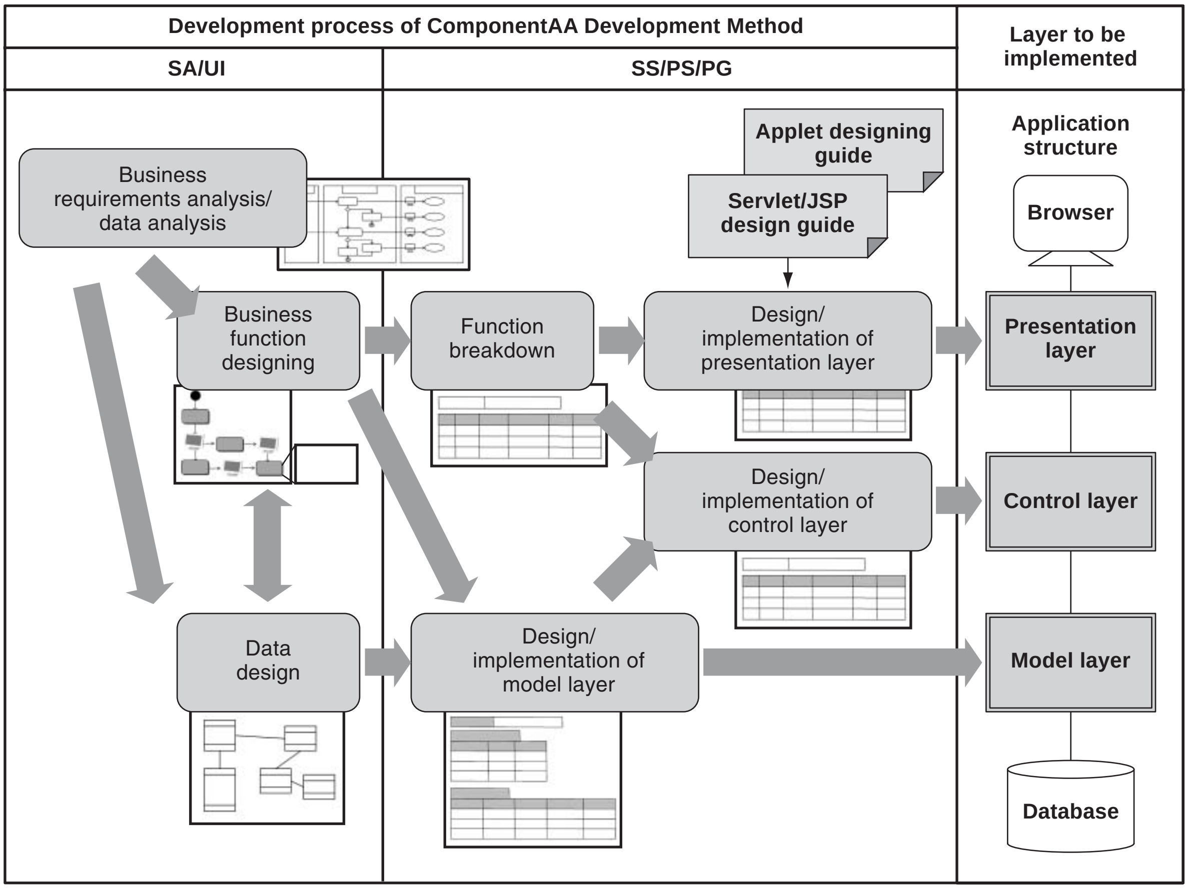

Development process of ComponentAA Development Method
SA/UI
SS/PS/PG
Layer to be implemented
Business requirements analysis/ data analysis
Business function designing
Data design
Function breakdown
Applet designing guide
Servlet/JSP design guide
Design/ implementation of presentation layer
Design/ implementation of control layer
Design/ implementation of model layer
Application structure
Browser
Presentation layer
Control layer
Model layer
Database

Figure 4
Development process corresponding to Web application architecture.

it should be written). By combining UML with ComponentAA Development Method, UML can be used very effectively.

3.3 Proven integration method of data analysis and designing technique

To develop an appropriate system for business requirements, data analysis and design techniques are effective. This is also true when developing Web applications implemented based on object-oriented technologies. It is necessary to fill the gap between data analysis and design with, for example, an entity relationship diagram (ER diagram) in the upper process and an implementation based on object-oriented technology. Object-oriented technologies enable great flexibility in design and implementation; consequently, without appropriate design guidelines, the application structure becomes inconsistent and the quality and maintainability of the system becomes low.

ComponentAA Development Method provides design guidelines that reflect data analysis and design in the implementation of components in the model layer, so data analysis and designing techniques, together with object-oriented implementation tasks, are integrated into a single development standard with no discrepancies.

3.4 Achievements and future plans

ComponentAA Development Method has been applied to more than 50 projects since its release in April 2004. Its document standard section is disclosed to the public through a Web site, and it has been downloaded and is being used by over 1080 people.

Web application technologies are constantly developing, and new implementation technologies and frameworks are continuously appearing, some of which are becoming popular. We will continuously enhance ComponentAA Development Method, focusing on Web applications, so it keeps up with new technologies and continue to provide leading-edge development standards.

4. Secure Web Application

As Web application development accelerates, attacks from the Internet and other external attacks such as injection attacks and fishing are sure to increase at the same time. **Figure 5** shows an SQL injection attack, which takes advantage of the Internet's vulnerability. In this example, "A0012' OR 'A'='A" is accidentally entered as a product code on a purchase information inquiry screen. The Web application interprets this as an SQL statement requesting a search of all items, and as a result, transaction information such information intended to be disclosed at a later date or pricing information for other customers is leaked. Although a high-quality, high-reliability information system has been developed, it may contain vulnerabilities that necessitate costly modifications or cause the service to stop, which may lead to trust issues.

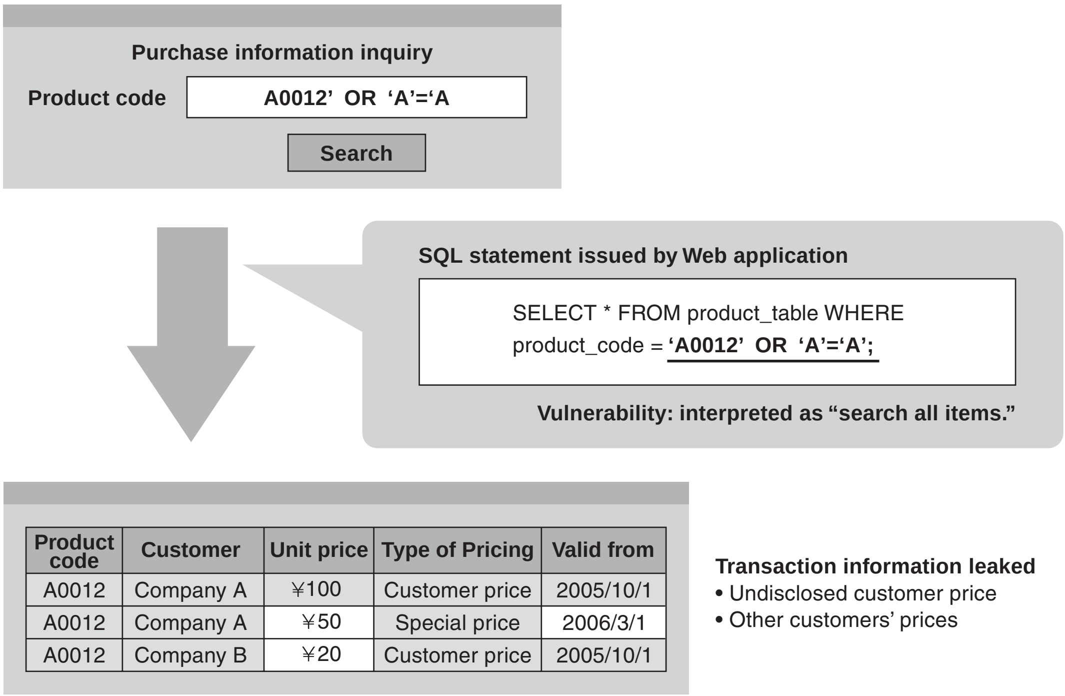
These problems illustrate that vulnerability countermeasures are becoming important in Web application development.[4),5)]

In this section, SDAS secure programming, which Fujitsu is working on, is explained.

4.1 Activation of vulnerability countermeasures

A predominant style of countermeasures often seen today is to assure quality by applying separate fee-based services. These services have been coping with these issues by analyzing vulnerability information and making full use of diagnostic tools. The vulnerability information is provided by the BugTraq mailing list from SecurityFocus and by the Open Web Application Security Project (OWASP).[6)]

In Japan in July 2004, the "Standards for Handling Software Vulnerability Information," was announced by the Ministry of Economy, Trade and Industry, and the IPA security center (IPA/ISEC) started receiving vulnerability-related reports about Web applications.[7)] In addition, vulnerability diagnosis services for Web applications have been started.


Purchase information inquiry
Product code
A0012' OR 'A'='A
Search
SQL statement issued by Web application
Vulnerability: interpreted as "search all items."
Transaction information leaked
• Undisclosed customer price
• Other customers' prices

```
SELECT * FROM product_table WHERE
product_code = 'A0012' OR 'A'='A';
```

Product code	Customer	Unit price	Type of Pricing	Valid from
A0012	Company A	¥100	Customer price	2005/10/1
A0012	Company A	¥50	Special price	2006/3/1
A0012	Company B	¥20	Customer price	2005/10/1

Figure 5
Example of SQL injection attack using vulnerability of Web application.

4.2 Challenges of vulnerability countermeasures

These services can provide fine-tuned countermeasures suited to a customer's system configuration by 1) limiting the diversification of vulnerabilities when Web application development technologies change and 2) providing threat analyses and ways of coping by fully exploiting know-how to find these vulnerabilities. However, these services tend to be expensive.

In many cases, these services are applied just before an application starts, and a system for implementing countermeasures during development has not yet been established.

It is therefore necessary to make arrangements based on the roles of developers. Also, the viewpoints of the team that develops business logic based on business specifications and the team that provides the processing method that makes the business logic function as a Web application should be considered when developing countermeasures. In the early stages, security countermeasures should be tackled by all teams, and they must be strictly controlled by considering who takes care of what. Otherwise, the countermeasures cannot be formalized and increases in costs and diagnosis time may occur.

4.3 Determining the type of countermeasures

As shown in **Figure 6**, SDAS provides a "Secure Web programming guide" for developing secure Web applications as part of the development standards. Mandatory countermeasures to be included in application development costs and the implementation of non-functional requirements — which must be considered as a mandatory requirement in the same manner as reliability and operational countermeasures — are handled separately. This separation enables the items to be included in application development costs to be discussed with the customer in advance. By establishing this baseline and systemizing already known vulnerability countermeasures

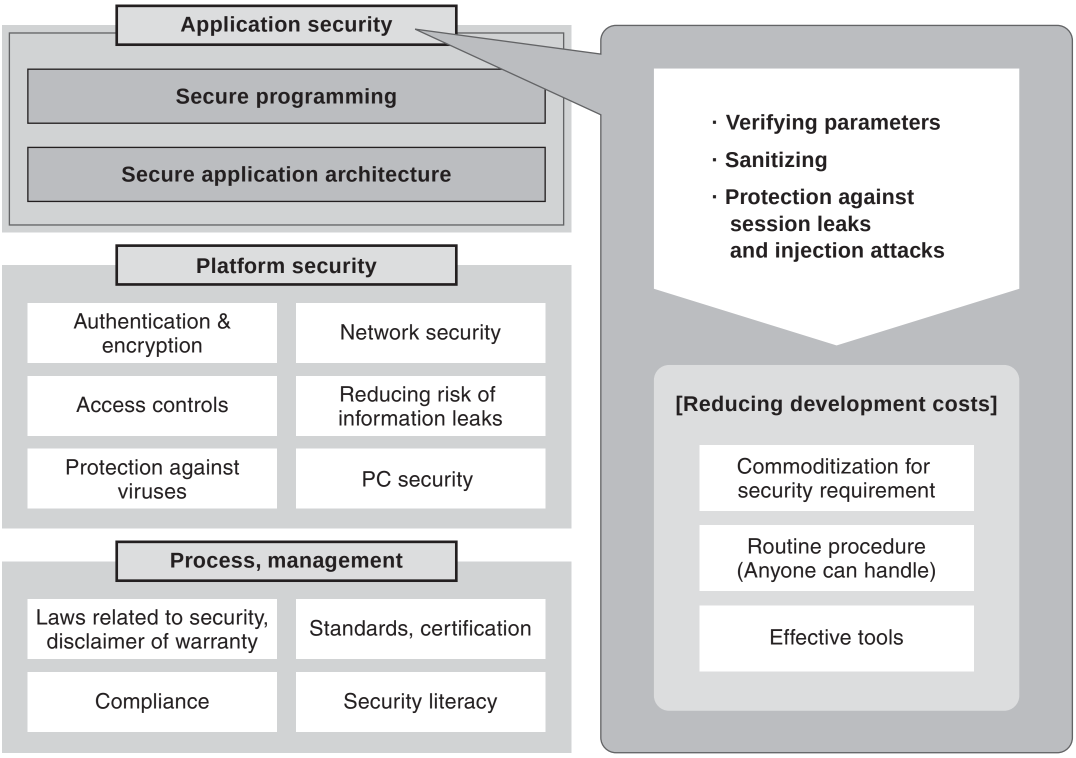

Application security
Secure programming
Secure application architecture
Platform security
Authentication & encryption
Network security
Access controls
Reducing risk of information leaks
Protection against viruses
PC security
Process, management
Laws related to security, disclaimer of warranty
Standards, certification
Compliance
Security literacy
· Verifying parameters
· Sanitizing
· Protection against session leaks and injection attacks
[Reducing development costs]
Commoditization for security requirement
Routine procedure (Anyone can handle)
Effective tools

Figure 6
Scope of SDAS Web application securities.

beforehand, additional vulnerability countermeasures that could become necessary on a daily basis can be distinguished from the already known ones, and it becomes possible to determine whether they should be included in the baseline or dealt with at extra expense.

It will also be important to consider the timing of countermeasures and process selection and ways to inspect things that must not be done. So far, third parties have performed these two tasks, resulting in high costs. Therefore, to reduce costs, the following three requirements will become important (**Figure 7**).

1) Confirmation methods that enable self-implementation and self-correction.
2) Detection tools for mass-production items.
3) Frameworks and application processing methods that have been verified in advance to realize 1) and 2).

4.4 Further determination of types

The concept of reducing costs by determining the types of countermeasures is a rather new one that requires extra work, and to promote it in system development projects, Fujitsu will use SDAS to further promote improvements in initial examinations in interviews with workers and diagnostic procedures.

Although the principles of vulnerability countermeasures are relatively stable, diagnostic methods and processing methods must be quickly developed to keep up with today's rapid changes in technology. We also aim to establish ways to handle cases that cannot be managed by the baseline, for example, authentications that require threat analysis or encryption countermeasures, as well as ways to accumulate proven processing methods and know-how and share information.

Category		Check points	Techniques
Business logic	Applet, Java code	35	Code checker: SIMPLIA/JF Kiyacker, etc.
	Servlet (JSP, JavaScript)	11	Diagnostic tool: String search, AppScan, etc.
Application architecture	• Authentication, SSO, encryption, session management, etc. • Development standard, etc.	16	Consulting (Including coverage range of frameworks)
Operation, platform configuration, etc.		13	

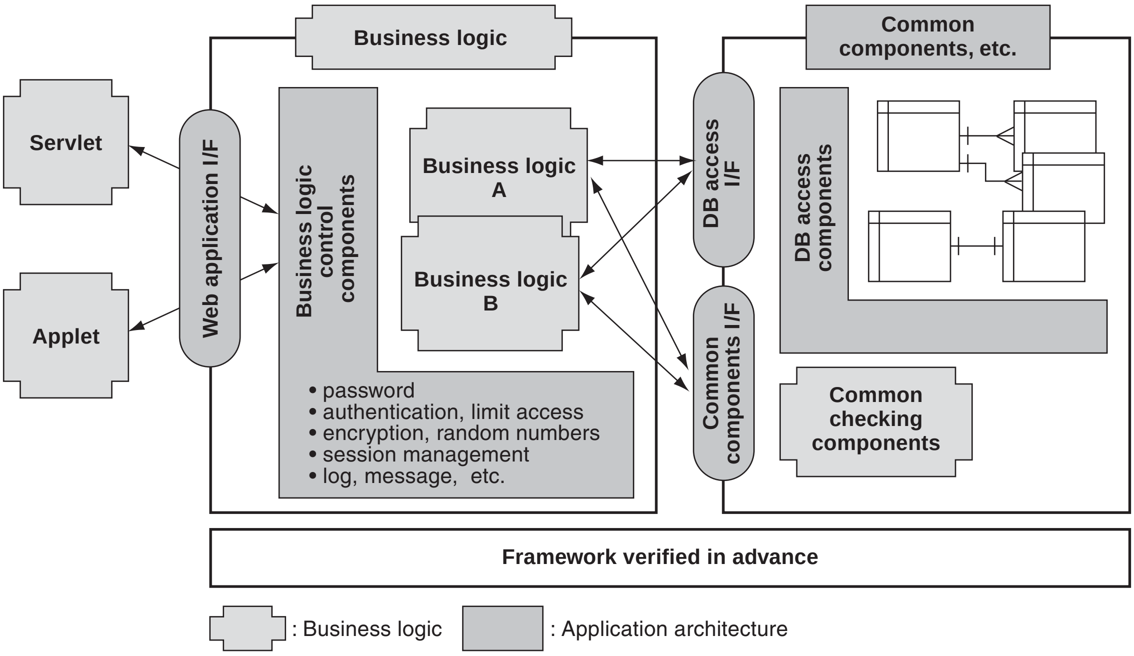


Figure 7
Protection profiles for vulnerabilities of Web applications.

5. Conclusion

This paper introduced SDEM21, which is a basic concept for system construction, and the SDEM practical template, which is a code of conduct for SEs. It also introduced the SDEM practical development standard, ComponentAA Development Method, and Secure Web programming guide, which are practical development standards that can be used for standardized tasks at a project site. We will continue to promote and enhance these standards to make projects more effective, taking into account the trends in new technologies.

References

1) Fujitsu: ComponentAA Development Method. (in Japanese).
http://segroup.fujitsu.com/sdas/technology/develop_guide/1_caa.html

2) EJB Component Consortium: Reusable EJB Component Design (technical report) DE-00-01. (in Japanese).
http://www.ejbcons.gr.jp/rules/DE-00-01-PR.PDF

3) Object Management Group: Superstructure Specification, v2.0 (formal/05-07-04UML).
http://www.omg.org/

4) H. Takagi: 40 rules for secure Web application program development. 2003. (in Japanese).
http://java-house.jp/~takagi/paper/idg-jwd2003-takagi-dist.pdf

5) IPA ISEC: Secure programming course. (in Japanese).
http://www.ipa.go.jp/security/awareness/vendor/programming/

6) OWASP: The Ten Most Critical Web Application Security Vulnerabilities.
http://www.owasp.org/

7) J. Hayakashi: About the handling of the security vulnerability information. (in Japanese), *Information Processing Society of Japan Magazine*, **46**, 6, p.662-671 (2005).

Ryoko Saito, *Fujitsu Ltd.*
Ms. Saito received the B.A. degree in Language and Culture from Tokyo University of Foreign Studies, Japan in 2002. She joined Fujitsu Ltd., Tokyo, Japan in 2002, where she has been engaged in development of standard processes for system engineers at Fujitsu.

Satoru Okiyama, *Fujitsu Ltd.*
Mr. Okiyama received the B.S. degree in Mathematics from Tokyo University of Science, Japan in 1989 and the M.S. degree in Mathematics from Nagoya University, Japan in 1991. He joined Fujitsu Ltd., Tokyo, Japan in 1992, where he has been engaged in development and application support of technology in the upper processes of Fujitsu's System Development Methodology, SDAS.

Fusami Hirai, *Fujitsu Ltd.*
Mr. Hirai received the B.S. degree in Science and Engineering from Aoyama Gakuin University, Japan in 1990. He joined Fujitsu Ltd., Japan in 1990, where he has been engaged in development and support of application architectures for industrial business systems.