# Requests and Main Problems Regarding Configuration Management in Open System Development — Management in Short-Term, Congested Developments —

● Masashi Iizuka

*(Manuscript received April 7, 2006)*

**To cope with the drastically changing business environment, enterprises are becoming strongly focused on ways to achieve speed-oriented management, improve management efficiency, and reduce costs, even at their system development sites. To help them achieve these goals, IT is becoming more open and is expanding its range of applications. For example, Fujitsu has developed configuration management models that target not only conventional library management areas, but also linkages with other management areas and areas of system specifications. We are currently verifying the effectiveness of these models in individual business fields. We regard problem-solving methods for configuration management as configuration management solutions. This paper describes the configuration management models we have developed that standardize these solutions and problem-solving methods by using configuration management solutions.**

## 1. Introduction

In today's business environment, international competition has intensified due to globalization and the removal of borders. In this situation, a faster pace of business, higher management efficiency, and lower costs are essential requirements for establishing a differentiation strategy that secures competitive superiority. To meet these requirements, Fujitsu's system integration (SI) division must speed up its system development and reduce its costs, and it is a shared mission between the customer and Fujitsu to quickly construct and operate stable, high-quality information technology (IT) systems with functions that meet requirements. In tandem with these business environment changes, IT has been moving in the open-source direction, and its technology area has become broader and more complicated. Against this backdrop, it has become necessary to do the following in project management at system development sites for large-scale business systems:

1) Initiate projects without clarifying requirement specifications in short-term projects
2) Respond to continuous specification changes and continue pursuing differentiation achieved by IT
3) Support the distributed, parallel, and congested developments that have become usual for open systems
4) Handle a geometrically increasing number of components in object-oriented developments
5) Support complicated software implementation structures caused by the incomplete definition of software structures peculiar to open-system languages

In configuration management, the requirements in 4) and 5) have added to broader and more

severe challenges.

In some cases, these challenges in configuration management have necessitated the implementation of extremely expensive "human wave" tactics, and the risks involved in developing large-scale, open mission-critical systems are becoming increasingly higher.

Therefore, to respond to these challenges, Fujitsu has built a configuration management model as a solution for configuration management and has been verifying it in various business fields.

This paper defines what requirements are needed in configuration management and proposes a configuration management model for overcoming various challenges. In the Conclusion, the current status and the future developments of the configuration management model are described

## 2. Concept of configuration management and requirements for its realization

Prior to modeling configuration management, its basic concept is defined as the life-cycle management (LCM) of incidents. In other words, configuration management in a broad sense is defined to ensure the resources derived from incidents are organized as services. **Figure 1** shows the conceptual scheme.

In today's SI business model, where speed, high efficiency, and high quality are required at the same time, the requirements for realizing the concept of configuration management described above are as follows.

1) In complex system development, multiple incidents are congested (a state in which incidents occur one after another before the current incident is completed), simultaneous parallel developments are ongoing, and the development period is short. In this type of development, configuration management must be adaptable to changes, always in control, and able to detect changes and quickly respond to them.
2) To optimize project development work in various models, methods, and environments for development, configuration management must, based on the characteristics of each project, ensure applicability so that the development process is consistent and the development style is adaptable.
3) Configuration management must realize functions for prompting the project leader to make quick decisions.

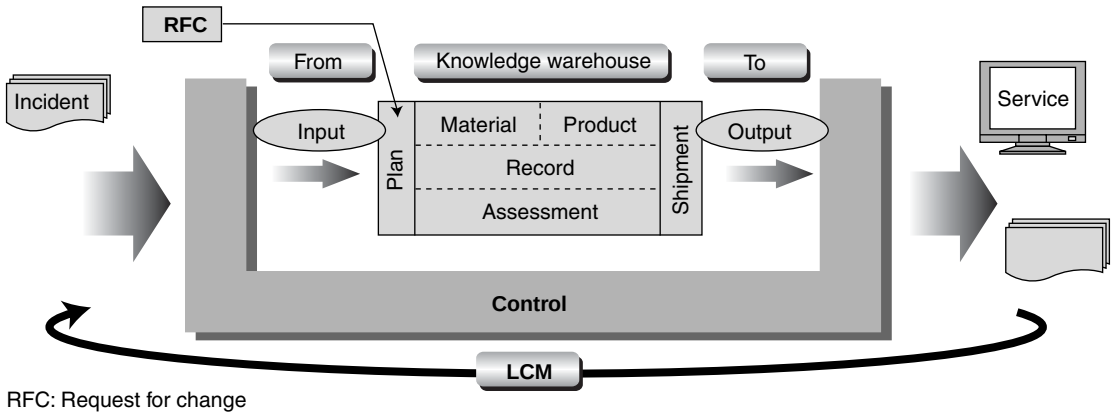


Figure 1
Concept of configuration management: life-cycle management of incident management.

## 3. Challenges in configuration management

The following challenges must be overcome to realize good configuration management.

1) Early development of a configuration management plan consistent with the project plan

To provide tolerance for changes and growing complexity and retain applicability for a variety of development styles, the configuration management plan must be integrated with the project plan in the planning phase of SI. Also, it is necessary to set up, as early as possible, configuration management functions for achieving overall optimization that considers the system development style, organizational planning, estimates, and other factors.

2) Clarification of configuration management areas

In today's short-term developments, for example, object-oriented development dealing with complex components, management organizations of multiple areas must cooperate with each other to clarify the overall scope of configuration management.

3) Structural model specifications and standardization of open software

In large-scale SI, where orders for goods from affiliated companies may be issued in the design phase, various development styles brought about by loosely structured open-system languages tend to result in disunited and disordered software structures due to differences among development cultures. This tendency leads to a notable reduction in management efficiency. For this reason, it is necessary to specify strict software structural models and integrate them through standardization.

4) Realization of powerful tools covering all the areas in LCM

In order to perform high-quality, efficient configuration management in short-term, congested system developments, it is essential to construct a configuration management system that consistently supports each LCM area from the initial design of the system to the maintenance area and the traditional library management area.

## 4. Configuration management model to overcome challenges

To overcome the challenges, we developed a configuration management model based on the following standardized policies.

### 4.1 Early development of configuration management plan

1) Process modeling of management areas

By defining the "Plan-Do-Check-Act" process as a process model in planning and operating configuration management, a clear roadmap to high-precision, early development of plans and appropriate operation can be created.

2) Development of configuration management plan templates

We have developed templates for a configuration management plan that follows the area definition of our configuration management solution. The templates consist of the following items that conform to policies advocated by the IT Infrastructure Library (ITIL)[1] and conform to the configuration management plan made public as an internal standard.

- The objective and achievement goal of configuration management
- Target areas for management and definition of applicable areas
- Organizational model for configuration management
- Definition of work breakdown structure (WBS)
- Development policy of configuration management plan
- Identification and tracking of specification clarity
- Instructions for handling the archive and cabinet
- Staff training materials

These templates improve completeness when adding areas applicable to the solution of the plan

and can be reused to improve efficiency for early development of the plan.

## 4.2 Clarification of scope definition in configuration management

By applying the concept of scope management, configuration management is clarified with the following three scope definitions.

1) Clear definition of management areas in light of the target areas and the applicable areas (scope definition 1: management area definition)

The concepts of the area definition of configuration management are as follows (**Figure 2**):

- To clarify which areas are management targets in the configuration management plan, we defined three target areas of configuration management: applications, infrastructure, and the development and test environment.
- By counterpoising the applicable areas of configuration management in the project management area against the SDAS development system, we have developed a model in which each phase of the project plan; estimates and organizational plan; design, development, and testing; and LCM/application portfolio management (APM) is mapped in the plan.

2) Clarification of the solution areas in configuration management (scope definition 2: solution area definition)

We defined each configuration management process in each area (e.g., business planning, design, production, testing, and LCM/APM) in WBS and clearly specified the solution areas in configuration management that require risk control in the planning and operational phase as a solution map.

Part of the solution map is shown in **Figure 3**.

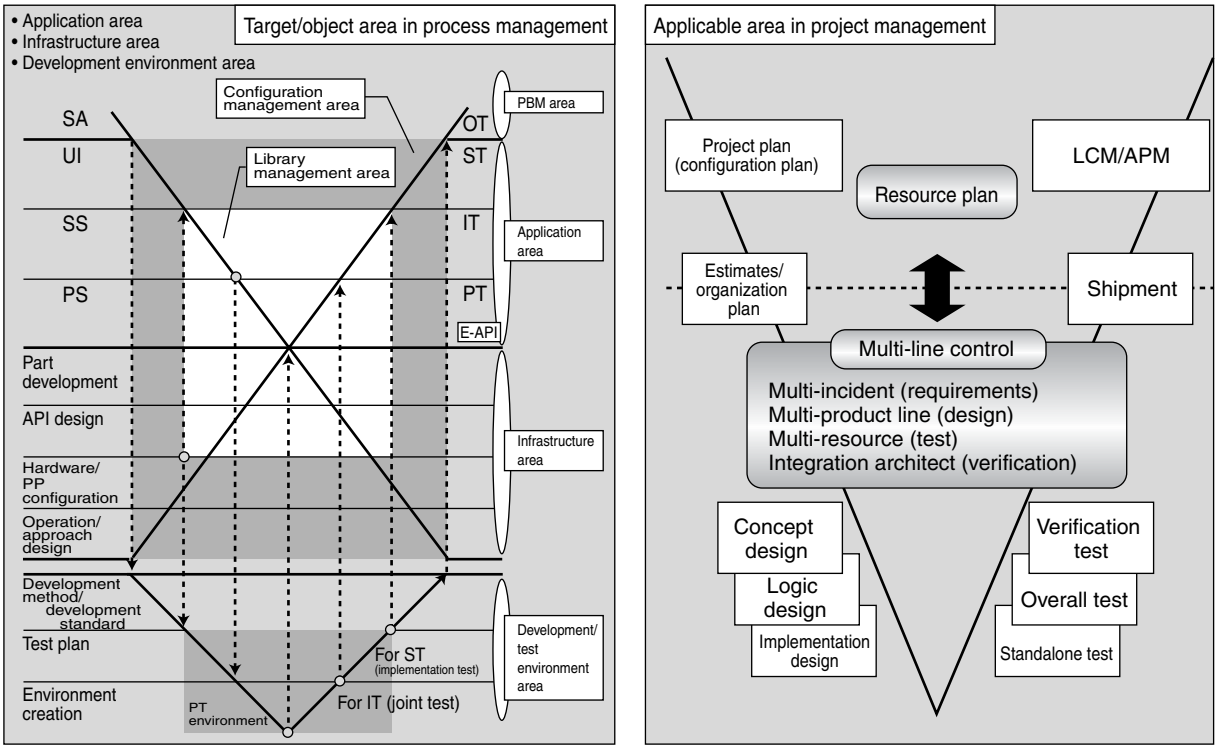


Figure 2
Definition area of configuration management.

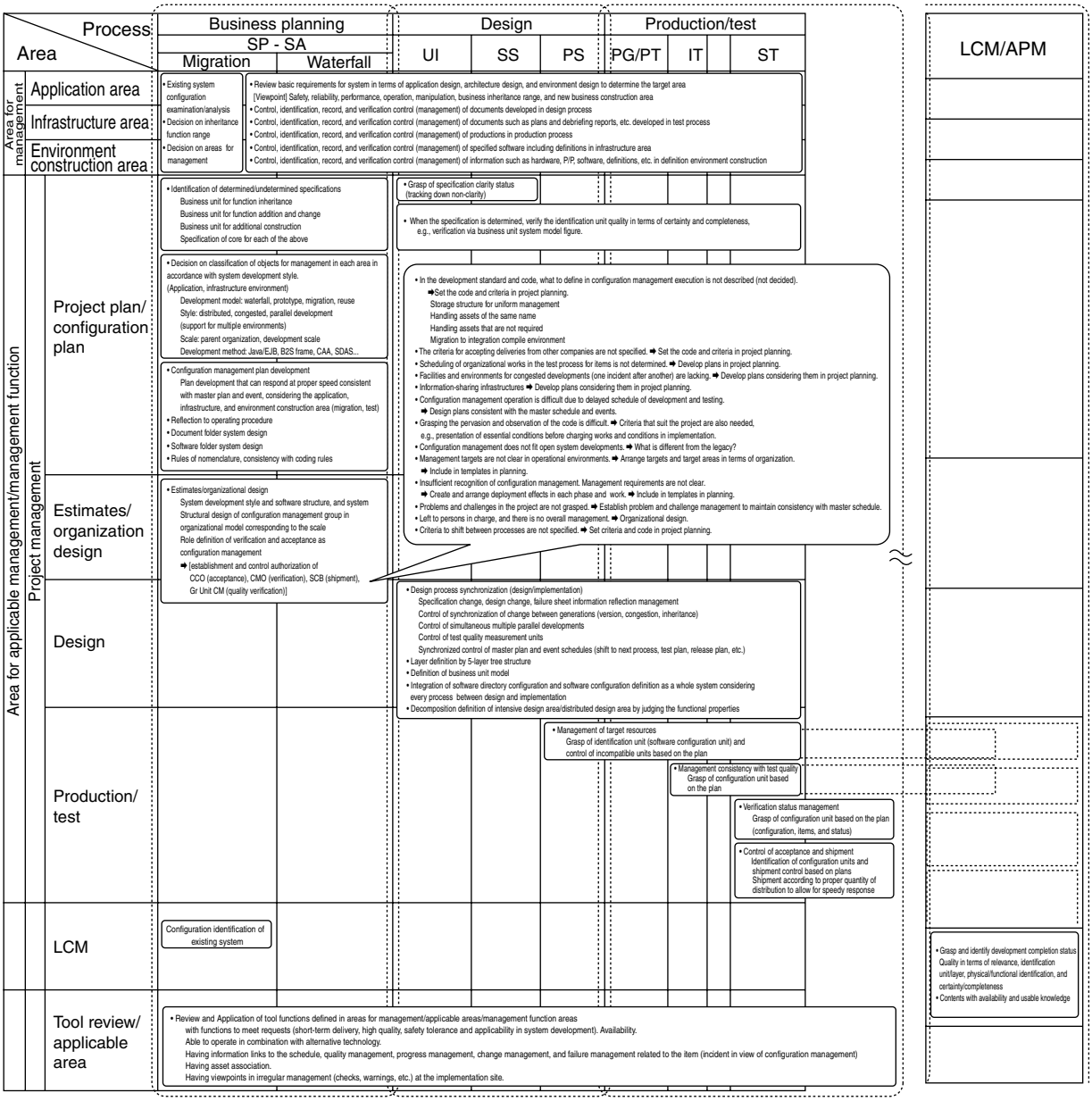


Figure 3
Example of solution map.

3) Development of a checklist for configuration management (scope definition 3: verification of the scope)

To enable an objective and quantitative assessment of whether configuration management of a process is properly planned, operated, and maintained, we developed a checklist. In contrast with the WBS area defined in the previous section, this list makes it possible to calculate validity scores in the planning, design and development, and LCM/APM phases. It can be used by a third party to monitor whether the operational quality and efficiency are established and maintained (i.e., perform configuration audits).

## 4.3 Tree diagram (tree structure) modeling of software structure

Mission-critical open systems still have a large amount of application software and software implementation logic (e.g., classes and methods)

peculiar to object-oriented languages that often fail to represent business layers as business systems. This can significantly reduce efficiency and quality in congested developments and highlights the importance of starting maintenance in the development phase. To solve this problem, focusing on the fact that the software structure of a large-scale business system is consistent with the organizational structure of the customer, we have defined and simplified the software structure as a five-layer tree-structure model and specified the storage structure of design documents and software materials as a five-layer folder model (**Figure 4**). By defining and identifying what is managed with a consistent tree structure regardless of the development style, information about the configuration, for example, its current status, can be identified for software change management.

## 4.4 Planning and design development of CMDB-centered configuration management system

When constructing a configuration management database (CMDB) advocated by ITIL, it must conform to the style of application. We therefore based our database model on the following.

1) Interoperation (improved applicability) with other configuration management product groups, including configuration management support tools developed by Fujitsu, by normalizing the configuration management components in the database and including a more standardized application program interface (API).
2) Tolerance for speed-focused development without clear specifications by making a database of complicated management components to identify and keep track of the

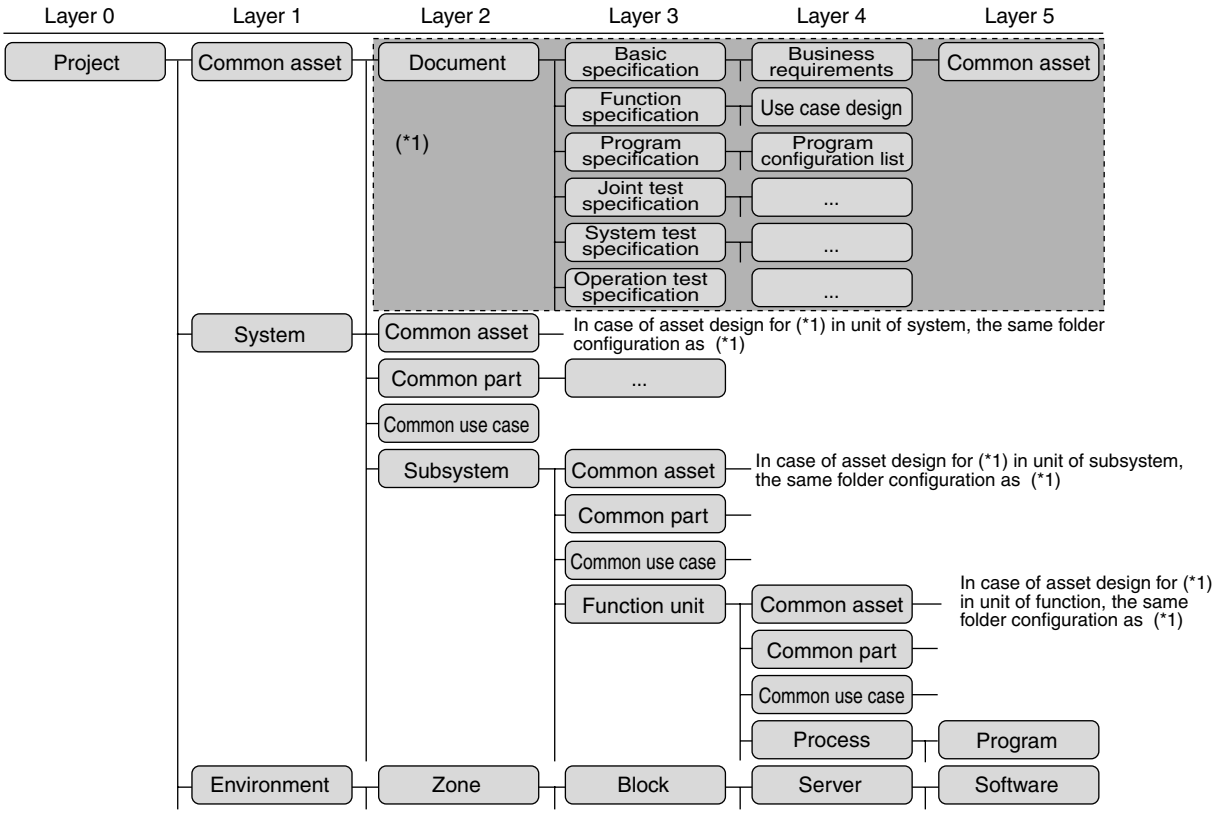


Figure 4
Five-layer tree model.

clarity of the specifications in the design document and software.

3) Close management cooperation with other project management areas such as process management, change management, and quality management for short-term, congested development by using an inter-database interface.

## 5. Effect of configuration management model and future development

In response to request for system developments with more complexity and less-clarified specifications caused by the need for differentiation and competitive superiority, we set the following objectives for the configuration management model approach described above.

1) Early development of the plan in order to clarify the configuration management risk at an early stage
2) Consistency with the system structured design that is central to system construction
3) Providing overall optimized services to development life-cycle management through modeling of the management process

To ensure that the configuration management model does not remain a mere desk plan, we have been verifying its efficacy in field activities.

### 5.1 Operation of configuration management WG by highly skilled staff

When developing this model, we consulted with 12 configuration management experts who have profound experience in overall project management (8 of these experts have been working in the WG for the last 10 months). By reviewing the configuration model repeatedly during these consultations, we realized our configuration management concept and improved the precision of documents, tools, and other items.

### 5.2 Verification of feasibility in model project

We verified the feasibility of our configuration management model by auditing the management status at project sites based on the configuration management checklist described in this paper. There are currently few applications of this model; however, we have already obtained statistically significant differences that indicate its efficacy in the verification process.

Therefore, we have concluded that significant differences can be verified, which indicates that the rationalization of management work based on the configuration management model leads to the promotion of efficiency in system management and operation. Further verifications will be performed to improve the precision of the model.

### 5.3 Collection and reflection of opinions from highly skilled architects

Through the present WG activities, we disclosed our model internally and then collected opinions from highly skilled architects and reflected their opinions in our study of configuration management.

### 5.4 Interoperation with other configuration management-related products developed by Fujitsu by inter-organizational cooperation

The groups in charge of the development and application promotion of the configuration management tool product SIMPLIA/SC-Manager[2] and an incident management ASP system developed by Fujitsu participated in this WG. We have reviewed the product interoperation with CMDB described above. We are planning to use our APM service to develop models and tools for configuration management. These models and tools will have a greater impact on quality in interoperation from incident management to configuration management, even for new development projects.

## 6. Conclusion

This paper described the challenges for configuration management based on today's architecture change, and introduced a configuration management model that Fujitsu has developed as a solution.

This model has been incorporated into the SDAS management system, allowing for organic interoperation with other software configuration management tools developed by Fujitsu.

We hope this paper will help you develop your own configuration management solutions.

## References

1) Office of Government Commerce (OGC): IT Infrastructure Library (ITIL).
*http://www.ogc.gov.uk/index.asp?id=2261*
2) Fujitsu: Web Site of SIMPLIA/SC-Manager. (in Japanese).
*http://software.fujitsu.com/jp/simplia/introduction/sc-manager_pc.html*

**Masashi Iizuka**, *Fujitsu Ltd.*
Mr. Iizuka received the B.E. degree in Electronic Engineering from Hokkaido University, Sapporo, Japan in 1980. He joined Fujitsu Ltd., Tokyo, Japan in 1980, where he was engaged in application program development and project management of systems dedicated for a telecommunication company for 20 years. Also, he has been engaged in support of application program development since 2003.