Information Technology Engineers Skill Standards

Technical Engineers (Embedded Systems)

Contents

Translated this Skill Standard（update on September 29, 2000） from Japanese into English on July 31, 2000

Japan Information Processing Development Corporation

Central Academy of Information Technology

1. Overview

1.1 Background of developing the “Information Technology Engineers Skill Standards”

At present, great hopes are placed on information technology as the sources of industry regeneration and new economic growth. This is because the roles of IT have been expanded from the tools for manufacturing cost reduction and service speedup to those for effective collaboration among enterprises and the creation of new industries. From now on, the rise or fall of an enterprise will be determined by quality of computerization investment. It is therefore an urgent matter to bring up engineers who construct advanced information systems and those who utilize them.

In view of this, the Central Academy of Information Technology has repeated a study on how to bring up, evaluate, and select good engineers who can show their practical ability on actual jobs. As a conclusion, the academy decided to establish the “information technology engineers skill standards” centering on the criteria to determine whether the required jobs can be performed adequately or not.

1.2 Significance and objective of developing the “Information Technology Engineers Skill Standards”

The results of surveys that the Central Academy of Information Technology has conducted on information processing engineers have suggested an important issue to be solved in the industrial world and by educational institutions such as schools. The issue is the establishment of the guidelines that clearly define what the industrial and educational worlds are expecting to get. While these guidelines need to define the level of knowledge, skills and capability to be equipped with by IT personnel (engineers) who do the actual jobs in the industrial world, they need to define the models of IT engineers who can be accepted internationally, and the ways how schools and other educational institutions should conduct education training on the basis of these models. One example of the guidelines is the “Skill Standard for IT Engineers” developed by the Northwest Center for Emerging Technologies (NWCET) as part of the establishment of “Skill Standards” by the US Department of Labor.

The “Information Technology Engineers Skill Standards” have been developed as a tool that solves the issue mentioned above, and apply to all the sections of the information technology engineers examinations as criteria to evaluate the skills of engineers who have been brought up. The application of this skill standard is significant for the industrial world in “recruiting human resources with the guaranteed ability to do actual jobs.” For educational institutions such as schools, this is significant for “understanding and confirming the knowledge, ability, and the achievement levels of the engineers required by enterprises.” For government agencies, this is significant for “grasping the technical level of the entire industrial world.”

1.3 Configuration of the "Information Technology Engineers Skill Standards"

The "Information Technology Engineers Skill Standards" is a tool that provides information about knowledge and skill needed to do jobs such as building, operational control, usage and evaluation of IT system in organizations such as corporations. It also provides indicators to determine the outcome of jobs. "Information Technology Engineers Examinations: Overview of the New System" and "Information Technology Engineers Examinations: Scope of Examinations" describe knowledge, technology (technical knowledge), and ability that information processing engineers need to have, and performance indicators (listed in 1), 2), and 3) below). The established skill standards describe these points more specifically by consulting actual jobs.

1) Roles and jobs
2) Expected technical levels
3) Scopes of examinations: examination in the morning and that in the afternoon
   (The above information can be downloaded to access http://www.jitec.jipdec.or.jp/.)

The "Information Technology Engineers Skill Standards" consists of three kinds of technical information described below. In this standard, individual skill standards are established for each examinees classified according to examination categories.

(1) Key activities

This chapter describes jobs that are keys unique to each examination categories. It describes the "roles and jobs" in 1) above more specifically.

(2) Skill criteria

This chapter describes what knowledge and skill should be used to do the key activities in (1) above, and also describe performance indicators to determine what outcome should be obtained. It describes "expected technical levels" in 2) above more specifically.

(3) Body of knowledge

This chapter systematically describes common knowledge independent of examination categories and knowledge needed to do the key activities in (1) above. This chapter also covers the "scopes of examinations" in 3) above.

1.4 Image of “technical engineers (embedded systems)” and skill standards

These skill standards have been drawn up by applying the framework of the Information Technology Engineers Skill Standards, which have so far been described, to “technical engineers (embedded systems).”

(1) Image of technical engineers (embedded systems)

In a typical embedded-system development project, technical engineers (embedded systems) engage in the formulation of a system development process, preparation of requirement specifications, and system design. In these basic jobs, they are required to have the ability to perform a series of operations such as functional design, detail design, preparation of a system development environment, system realization, and integrated tests covering both hardware and software.

In addition, before starting system development, they are expected to play the role of evaluating the technical and economic effects of the software, microprocessors, system LSI, and other components to be incorporated into products.

(2) Skill standard

The following skill standards apply to technical engineers (embedded systems):

1) IT common body of knowledge
2) Technical engineer (embedded system) skill standards
   - Key activities, skill standards, practical body of knowledge, and core body of knowledge

2. Key Activities

Key activities in an embedded-system development project refer to procedural items described about operations in the system development phase, which is the basic job area for technical engineers (embedded systems). In this skill standard, the above job area is called an "embedded-system development job process."

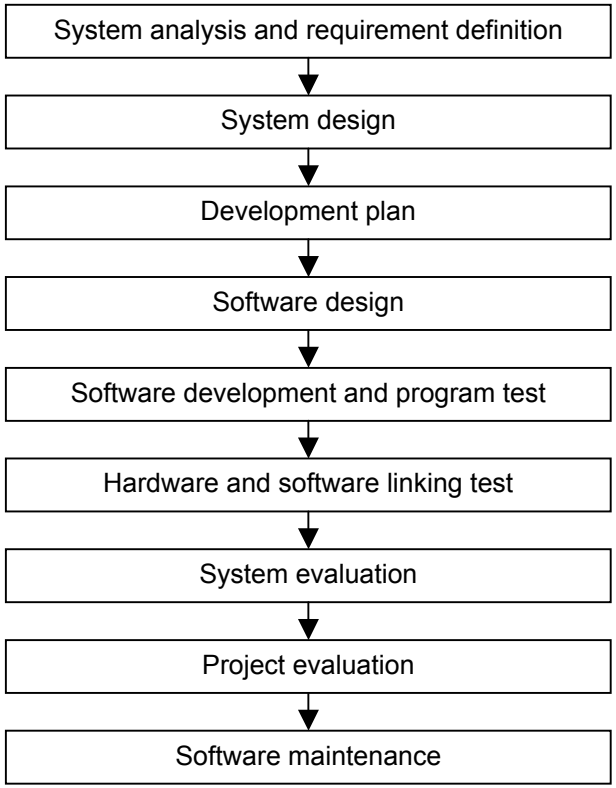

Figure 2-1 Embedded-system development job process

As shown in Figure 2-1, jobs in the embedded-system development job process are broken down into nine basic "activities."

Each activity is further broken down into detailed jobs called "tasks." This skill standard presents the embedded-system development job process in the following format:

Activity	Task	Job outline
1. Act 1	1-1 Task 1	x x x x x x x x x x x x x x
	1-2 Task 2	x x x x x x x x x x x x x
	1-3 Task 3	x x x x x x x x x x
2. Act 2	2-1 Task 1	x x x x x x x x x x x
	2-2 Task 2	x x x x x x x x x x x x x
	2-3 Task 3	x x x x x x x x x x x x x x
	2-4 Task 4	x x x x x x x x x x x x

Technical engineers (embedded systems) mainly take charge of the activities of "system analysis and requirement definition," "system design," and "development plan" shown in Figure 2-1. Furthermore, in view of the expected technical levels of technical engineers (embedded systems) described in "Information Technology Engineers Examinations: Overview of the New System," they are supposed to have the ability to take charge of the activities ranging from "software design" to "software maintenance" and also to lead the system development team.

[Embedded-system development job process]

Activity	Task	Job outline	
1. System analysis and requirement definition	1-1 Acquire and adjust requirements	(1) Understand the field in question (2) Hear from the customers, followed by confirmation and adjustment (3) Carry out market research (4) Grasp related laws, regulations, and standards	Acquire system requirements by understanding the field in question (including the preparation of a glossary and an inquiry into related laws and regulations), hearing from the customers (the scope of systematization, confirmation of decisions, and adjustment), and carrying out market research (substantiating the requirements and inquiring into market needs).
	1-2 Prepare work plans	(1) Prepare system analysis and requirement definition plans	Prepare concrete plans for implementing system analysis and requirement definition. In this process, determine analysis techniques, development standards, and development tools in consideration of such factors as the field, scale, past development record, and novelty.
	1-3 System analysis and requirement definition	(1) Systematically analyze and define functional requirements (2) Analyze and define network conditions and external interface specifications (3) Analyze and define human interface specifications (4) Analyze and define performance conditions (5) Analyze and define reliability and safety (6) Analyze and define maintenance conditions (7) Analyze and define other restrictive conditions (such as hardware) (8) Analyze feasibility (9) Analyze risks (10)Review the contents of requirement definitions	Arrange the customer requirements systematically by analyzing the functions and conditions required of the system. In the arrangement process, include not only functional requirements but also hardware restrictions and the like in the customer requirements, and arrange risk analysis for the requirements as well. In addition, decide whether and how to update embedded-system programs, as by debugging them and introducing new versions. Thus clarify the scope of development.

Activity	Task	Job outline	
	1-4 Survey related technologies and other companies' intellectual property rights	(1) Perform general technical surveys in a new field (2) Survey trends in hardware/software configurations and element technologies (3) Survey trends in development technology, environment, and tools (4) Inquire into other companies' technologies (5) Consider how to deal with other companies' technologies (6) Check for product liability problems	Survey the related technologies required to meet system requirements. Also check whether the system requirements or the means of meeting the system requirements pose any problems in connection with other companies' intellectual property rights or product liability.
	1-5 Study trade-offs between cost, period, and functional volume	(1) Estimate work volume and period (2) Estimate resource volume and cost (3) Scrutinize required items (4) Collect and analyze a track record of similar projects (5) Coordinate and negotiate with the customers (6) Reflect in required items	Prepare an estimate, considering the work volume and period, the resource volume and cost, the required delivery date and cost (budget), the functional volume (including restrictive conditions), and so forth. At this time, collect and analyze a track record of the development of similar projects as necessary in order to improve the precision of estimation. If the customers do not consent to the estimate, coordinate and negotiate with them.
	1-6 Summarize system specifications	(1) Finalize system specifications (2) Formulate design policies (3) Study the possibility of reuse of existing property (4) Clarify how to handle uncertain factors	Arrange the customers' requirements systematically and document them as system specifications, covering design policies (design techniques, proposals about the architecture to be read, design restrictions, and so on), the study of the possibility of reuse of existing property, and the impact of uncertain factors.
	1-7 Maintain work plans	(1) Execute system analysis and requirement definition work control	Schedule the use of resources to execute system analysis and requirement definition work; check whether the work is being performed to meet their objectives; and complete the work as planned.

Activity	Task	Job outline	
	1-8 Control work results and retain records	(1) Perform progress control and retain a track record of progress (2) Perform quality control and retain a quality record (3) Retain a record of access to other companies' data	Clearly grasp the current position in the work plans, detect and analyze problems early, and take measures. Control progress so that work will be performed in accordance with the development procedures. Also retain various track records.
	1-9 Review system analysis and requirement definitions	(1) Plan a review (2) Hold a review (3) Deal with items pointed out in a review (4) Judge whether system analysis and requirement definition work have been completed	Plan and hold a review of system specifications and track records (time, selection of reviewers, place, and so on), and take measures to deal with any items pointed out in the review. Based on the results of the review, judge whether the system analysis and the requirement definition work have been completed.
	1-10 Prepare an outline of a project plan	(1) Prepare an outline of a project plan	Prepare projections and guidelines concerning the future implementation of the project as an outline of a project plan.

Activity	Task	Job outline	
2. System design	2-1 Prepare a work plan	(1) Prepare a system design work plan	Prepare a plan specifying and detailing work items and performers in the system design stage. Clarify the division of responsibilities between hardware and software engineers. Determine design techniques, considering such factors as the field, scale, past development record, and novelty.
	2-2 Determine the hardware configuration of the system	(1) Understand system specifications (2) Determine system component devices (3) Determine devices to share the functions of interfacing external equipment (4) Determine a communication method between devices (5) Determine the internal configurations of devices (6) Determine inter-unit interfacing methods (7) Evaluate making system LSI	Study and determine the devices that constitute the system under system specifications as well as units (minimum system components with interfaces), and study and determine the interface, communication method, and so on between them. Also evaluate and determine whether system LSI should be made.
	2-3 Assignment of system functions to component devices	(1) Add functions for divided arrangement of required functions and evaluate them (2) Add initialization functions and evaluate them (3) Add RAS (Reliability, Availability and Serviceability) functions and evaluate them (4) Add multiplexing control functions (processors and communication paths) and evaluate them (5) Evaluate the interfacing methods between devices and between units (6) Define the interfaces between devices and between units	Along with the study of the hardware configuration, appropriately assign the system requirement functions to components and units, and evaluate them.

Activity	Task	Job outline	
	2-4 Determine the division of functions and performance between hardware and software	(1) Divide functions between hardware and software (2) Define hardware-software interface specifications (3) Divide the means of achieving system performance requirements (4) Select microprocessors to reason hardware function specifications and evaluate the making of system LSI	Determine an optimum division between the functions to be provided by the hardware of individual devices and units and the functions to be performed by software. Study interfaces between hardware and software; clarify the contacts between hardware and software; and define them in such a way as to leave no ambiguity about the interpretation of control signals, data, timing, and so on or no inadequacy of information. Determine an optimum division between hardware and software by studying not only functions but also the performance required of the system.
	2-5 Verify feasibility and perform a design review	(1) Verify feasibility by experiment (2) Perform a design review	Verify feasibility by experiment as needed. Also perform a design review and evaluation of feasibility from the aspects of both hardware and software.
	2-6 Summarize software specifications	(1) Finalize software requirement specifications (2) Determine the items to be described in software specifications (3) Clarify software design conditions (4) Clarify uncertain factors	Summarize software specifications for each device and unit (the unit of installation of software) and document them as software specifications.

Activity	Task	Job outline	
	2-7 Determine period, cost, and functional volume	(1) Estimate work items, work volume, period, development resources, and cost (2) Scrutinize balance between development period, cost, and functional volume	Estimate work items, work volume, period, development resources, and cost, considering hardware development. Also verify the propriety of the estimate by scrutinizing the balance between the development period, cost, and functional volume (including restrictive conditions).
	2-8 Maintain work plan	(1) Perform system design work control	Maintain and control the theme so that system design will be completed as planned.
	2-9 Control work results and retain records	(1) Perform progress control and retain a track record of progress (2) Perform quality control and retain a quality record (3) Retain a record of access to other companies' data	Clearly grasp the current position in the work plans, detect and analyze problems early, and take measures. Control progress so that work will be performed in accordance with the development procedures. Also retain various track records.
	2-10 Review system design	(1) Plan a review (2) Hold a review (3) Deal with items pointed out in a review (4) Judge whether system design has been completed	Plan and hold a review of software specifications and track records (time, selection of reviewers, place, and so on), and take measures to deal with any items pointed out in the review. Based on the results of the review, judge whether the system design has been completed.

Activity	Task	Job outline	
3. Development plan	3-1 Process plan	(1) Prepare a process plan (2) Prepare an estimate (3) Prepare a process control plan (4) Prepare an education plan	Prepare concrete plans to execute software development. In the case of embedded-system development, prepare the plan, emphasizing its consistency with the hardware development plan.
	3-2 Development environment preparation plan	(1) Make a plan to prepare a development tool environment (2) Make a plan to prepare development equipment (3) Prepare a standardization plan	Make a concrete plan to prepare a development environment. Prepare concrete plans concerning distributed development environments, work equipment, test equipment, and so on.
	3-3 Review plan	(1) Prepare a review plan	Prepare a concrete plan to perform a review.
	3-4 Hardware-software linking test plan	(1) Set the time to start linking tests (2) Prepare a test location plan (3) Make a test preparation plan (4) Prepare a personnel plan	Plan the conditions for conducting hardware-software linking tests, and prepare a concrete plan to conduct the tests.
	3-5 Embedded-system quality assurance plan	(1) Prepare a quality target plan (2) Prepare a quality control plan	Prepare a concrete plan to achieve quality targets and a mechanism to collect and utilize information.

Note 1: Process plan: Prepare a process, and estimate the scale and man-hours.

Note 2: Process control plan: A plan to control a process. Plan process control cycles and method, the method of grasping progress, the handling of control data, and so on.

Note 3: Quality target plan: Plan quality targets concerning the debugging ratio and so on.

Note 4: Quality control targets: A plan to control quality. Plan the collection of quality information and the utilization method.

Activity	Task	Job outline	
	3-6 Plan conferences with the hardware group and review the development plan	(1) Prepare plans for conferences with the hardware group (2) Review the development plan	Prepare plans for conferences with the hardware group. Also determine in advance the standard items (progress, specifications, quality, and so on) to be studied in the conferences. If it is judged impossible to carry through the development plan initially prepared from the aspects of hardware and software, coordinate with the divisions concerned and review and modify the development plan.
	3-7 Configuration control plan	(1) Extract documents subject to configuration control (2) Formulate configuration control operation rules	Prepare a concrete plan to realize configuration control and configuration control rules.
	3-8 Maintenance plan	(1) Prepare a maintenance plan	Define maintenance following system development, determine maintenance conditions, and prepare a maintenance plan.
	3-9 Prepare a written development plan	(1) Prepare a written development plan	Prepare a written software development plan.

Activity	Task	Job outline	
4. Software design	4-1 Prepare a work plan	(1) Concretize a work plan prepared in development planning (2) Identify work items and assign responsible workers (3) Prepare a design review execution plan	Based on the plan prepared in development planning, make a concrete plan to perform software design.
	4-2 Determine the software configuration	(1) Design the task configuration (2) Threads (3) Design common data (4) Define operating system resources (5) Design clusters (6) Design the module configuration (7) Allocate memory	Design software component elements, including the task configuration, threads, common data, operating system resources, classes, and module configuration. With respect to allocation to memories such as mask ROM and flash memory, consider the speed, capacity, and reliability of embedded-system programs together with the presence or absence of alterations. In the software design work, use the design techniques adopted in the project.
	4-3 Perform a design review	(1) Confirm the method of executing each job and its content (2) Confirm the realization of functions (services) (3) Evaluate task configuration design (4) Evaluate class design (5) Evaluate module configuration design	Perform a design review of the outcome of software design, and instruct rework as necessary.
	4-4 Summarize software design	(1) Prepare software design sheets	Consistently summarize the results of the design of the task configuration, threads, common data, operating system resources, classes, and module configuration, and document them as software design sheets.
	4-5 Maintain the work plan	(1) Perform software design work control (2) Control progress on a monthly and a weekly basis	Maintain and control the theme so that software design will be completed as planned.

Activity	Task	Job outline	
	4-6 Align and coordinate with the hardware group	(1) Hold a coordination conference with the hardware group (2) Prepare a rework plan (3) Execute rework	Hold a conference in accordance with the conference plan with the hardware group prepared in development planning, and align and coordinate progress, specifications, and quality. If any rework occurs with respect to software specifications, prepare a rework plan and redo the software design.
	4-7 Control work results and retain records	(1) Perform progress control and retain a track record of progress (2) Perform quality control and retain a quality record (3) Record the configuration of products and version number changes (4) Retain a record of access to other companies' data	Clearly grasp the current position in the work plans, detect and analyze problems early, and take measures. Also retain various track records, covering events ranging from modifications of specifications to the occurrence and disposal of problems and bugs.
	4-8 Review software design	(1) Plan a review (2) Hold a review (3) Deal with items pointed out in a review (4) Judge whether software design has been completed	Plan and hold a review of the products of software design and track records (time, selection of reviewers, place, and so on). Take measures to deal with any items pointed out in the review. Based on the results of the review, judge whether the software design has been completed.

Activity	Task	Job outline	
5. Create and test programs	5-1 Prepare a work plan	(1) Concretize the work plan prepared in development planning (2) Assign personnel, facilities, and tools (3) Plan progress and quality control methods (4) Prepare a detailed work plan (5) Extract the items to be controlled	Based on the plan prepared in development planning, make concrete plans to create programs and to test them.
	5-2 Create programs and extract program test items	(1) Determine the standard format for describing source programs (2) Prepare a programming environment (3) Create programs (4) Select test methods (5) Extract program unit-test items (6) Extract program test items (7) Determine fault isolation procedures	For programming, determine the standard format for describing source programs. Also select the test methods based on the quality levels required of programs, and extract the tests to be performed.
	5-3 Perform a code review and a design review of program test items	(1) Perform a code review (2) Review test methods and test items (3) Prepare a review check list (4) Report on the results of the review	Set a target number of bugs to be extracted in reviews, and perform reviews to improve the quality of the programs created and of the test methods and items extracted.
	5-4 Summarize program quality data	(1) Summarize programming	Summarize, evaluate, and record the program size, development period, comment ratio, test items, and the contents and the number of matters pointed out in the design review.

Activity	Task	Job outline	
	5-5 Perform program tests	(1) Build and maintain a program unit-test environment (2) Perform program unit tests (3) Edit the system (4) Build and maintain a program test environment (5) Perform program tests	Build an environment necessary for program tests and formulate rules. Next, perform program tests in accordance with the program test methods and items.
	5-6 Summarize program test results	(1) Summarize program unit tests (2) Summarize program tests	Summarize, evaluate, and record the test results, including the test period, number of tests performed, and number and contents of bugs detected. Based on the program test results, record whether quality has been built in as required.
	5-7 Maintain the work plan	(1) Perform progress and quality control (2) Confirm progress in hardware development (3) Carry out the detailed work plan	Maintain and control the theme so that programming and program tests will be completed as planned.
	5-8 Align and coordinate with the hardware group	(1) Hold a coordination conference with the hardware group (2) Prepare a rework plan (3) Execute rework	Hold a conference in accordance with the conference plan with the hardware group prepared in development planning, and align and coordinate progress, specifications, and quality. If any rework occurs with respect to programs, prepare a rework plan and redo the steps from programming to program tests. Amend higher level specifications as necessary.
	5-9 Control work results and retain records	(1) Perform progress control and retain a track record of progress (2) Perform quality control and retain a quality record (3) Control the history of program alterations (4) Make and retain a record of the root causes of bugs and the methods of dealing with them (5) Make a record of access to other companies' data	Clearly grasp the current position in the work plans, detect and analyze problems early, and take measures. Also retain a record of modifications of specifications, occurrences of problems and bugs, and measures taken to deal with them.

Activity	Task	Job outline	
	5-10 Review programming and program tests	(1) Plan a review (2) Hold a review (3) Deal with items pointed out in a review (4) Judge whether programming has been completed (5) Judge whether program tests have been completed	Plan and hold a review of the programming, program test results, and track records (time, selection of reviewers, place, and so on), and take measures to deal with any items pointed out in the review. Based on the results of the review, judge whether the programming and program tests have been completed.

Activity	Task	Job outline	
6. Hardware-software linking tests	6-1 Prepare a work plan	(1) Concretize the work plan prepared in development planning (2) Prepare a test personnel assignment plan (3) Plan the assignment of the order of using the test environment (4) Prepare a detailed work plan (5) Extract the items to be controlled	Based on the plan prepared in development planning, make a concrete plan to conduct hardware-software linking tests.
	6-2 Build a test environment	(1) Prepare test equipment and materials (2) Secure a place to conduct tests (3) Secure test personnel	Confirm the number of debugging tools, test personnel, and the place to deploy the system, and build a test environment.
	6-3 Extract test items, determine test procedures, and review them	(1) Clarify the scope and contents of tests (2) Formulate a linking test strategy (3) Determine the order of conducting tests (4) Determine the means of testing (5) Extract test items (6) Set test criteria (7) Prepare a test plan (8) Review linking test specifications	Prepare linking test specifications, extracting test items and determining test procedures. Review the prepared linking test specifications, and modify any matters pointed out.
	6-4 Conduct tests	(1) Educate development project members (2) Build and maintain a test environment (3) Conduct tests (4) Negotiate with other divisions (5) Deal with contingencies (6) Isolate faults (7) Conduct a confirmation test on the final version	Conduct the tests in accordance with the test specifications. Formulate and carry out measures to deal with bugs occurring in the tests.
	6-5 Maintain a work plan	(1) Perform progress and quality control (2) Carry out the detailed work plan	Maintain and control the test objectives so that the hardware-software linking tests will be completed as planned.

Activity	Task	Job outline	
	6-6 Align and coordinate with the hardware group	(1) Hold a coordination conference with the hardware group (2) Prepare a rework plan (3) Execute rework	Hold a conference in accordance with the conference plan with the hardware group prepared in development planning, and align and coordinate progress, specifications, and quality. If any rework occurs with respect to programs, prepare a rework plan and redo the steps from programming to linking tests. Amend higher level specifications as necessary.
	6-7 Control work results and retain records	(1) Perform progress control and retain a track record of progress (2) Perform quality control and retain a quality record (3) Control the history of program alterations (4) Make and retain a record of the root causes of bugs and the methods of dealing with them	Clearly grasp the current position in the work plans, detect and analyze problems early, and take measures. At this time, retain a record of modifications of specifications, occurrences of problems and bugs, and measures taken to deal with them, and also retain the minutes of overall progress conferences and so on as records.
	6-8 Review linking tests	(1) Plan a review (2) Hold a review (3) Deal with items pointed out in a review (4) Judge whether linking test have been completed	Plan and hold a review of the hardware-software linking tests and track records (time, selection of reviewers, place, and so on). Take measures to deal with any items pointed out in the review. Based on the results of the review, judge whether the hardware-software linking tests have been completed.

Activity	Task	Job outline	
7. System evaluation	7-1 Prepare a plan to receive an examination	(1) Prepare a plan to receive inspection	Prepare a concrete plan to receive inspection (arranging the products and the results of control).
	7-2 Prepare to receive inspection	(1) Arrange the products at each stage of the process (2) Arrange the results of control at each stage of the process (3) Request inspection	Put together the materials (plans, design sheets, records, and so on) that can prove that the functions and the like required of the system have been realized in the system. Next, request the inspection division or inspectors to perform inspection.
	7-3 Evaluate the system	(1) Analyze inspection results and take measures (2) Determine system evaluation items and criteria (3) Carry out system evaluation	Examine whether the developed system has the functions initially specified. Also evaluate the manuals attached to the system, the degree of achievement of technical targets, intellectual property rights, quality targets, and so forth.
	7-4 Transfer the system	(1) Prepare for transfer (2) Carry out transfer	Prepare the documents and other materials necessary for transfer. Transfer them together with the system to the shipment division.

Activity	Task	Job outline	
8. Evaluate the project	8-1 Evaluate the project	(1) Determine project evaluation items and criteria (2) Carry out project evaluation	Analyze the project from the viewpoint of QCD, and clarify the causes of success or failure.

Activity	Task	Job outline	
9. Maintain software	9-1 Prepare a maintenance plan	(1) Draw up a maintenance plan	Based on the plan prepared in development planning, draw up a concrete plan to carry out maintenance. Draw up the plan by extracting and studying the items and contents of work in accordance with the contents of maintenance (modification of functions, handling of complaints, and so on).
	9-2 Arrange development information for maintenance	(1) Arrange specifications (2) Arrange design sheets (3) Retain programs (4) Retain development records (5) Arrange data on the method of operating the development environment (6) Retain a record of revisions	Arrange and retain maintenance documents necessary for carrying out maintenance.
	9-3 Prepare a maintenance environment	(1) Prepare a version number control system (2) Prepare a customer information control system (3) Prepare a maintenance environment	Build an environment necessary for maintenance (version number control system, customer information control system, maintenance development environment).
	9-4 Carry out maintenance	(1) Instruct maintenance work (2) Carry out maintenance (fault correction, functional improvement) (3) Perform alteration control and complaint control (4) Reflect in the next project	Carry out work to correct faults and improve functions. Record alterations and complaints, and utilize them. Also analyze the complaints and reflect the results in the next project.
	9-5 Transfer the system	(1) Prepare for transfer (2) Carry out transfer	Prepare the documents and other materials necessary for transfer. Transfer them together with the system to the shipment division. Prepare documents, as necessary, describing the methods of maintenance (update of system programs).

3. Skill Criteria

The skill criteria correspond to tools (tables) that provide indicators to check the status of achievement of the embedded-system development job process described in the key activities. With these criteria, it is determined whether technical engineers (embedded systems) have promoted a series of jobs successfully according to proper sequence and by using proper techniques, proper knowledge, and proper skill.

The skill criteria provide indicators to indicate what outcome needs to be obtained ("performance indicators") as a result of job execution for each task of each of the nine activities. They also provide knowledge ("required knowledge") and skill ("required skill") required to do the jobs.

[Technical engineers (embedded systems) skill criteria]

1. System analysis and requirement definition				
No.	**Task**	**Performance indicators**	**Required knowledge**	**Required skill**
1-1	Acquire and adjust requirements	• User information is properly judged, and the users' conditions and problems are grasped accurately. • The requirements of the system are clarified, and a consensus is formed with the users on restrictions and so forth.	• Knowledge about the terminology, products, and technology in the users' industry • Knowledge about problem analysis techniques • Knowledge about marketing, including market trends and products • Knowledge about related laws, regulations, and standards	• Ability to obtain information from the users and sort it out • Ability concerning interviewing techniques • Ability to precisely identify the users' problems • Ability to collect market and other information • Ability to analyze and relate collected information • Ability to negotiate to adjust requirements • Ability to propose and present solutions to the users
1-2	Prepare a work plan	• Work concerning system analysis and requirement definition is all extracted. • An effective work plan is drawn up based on resources. • A consensus is reached with the users with respect to a work plan and risks. • The objects and time of a review and participating members are planned.	• Knowledge about techniques and work procedures in system analysis • Knowledge about resource allocation • Knowledge about project planning and control techniques	• Ability to extract work items and arrange them systematically based on the system analysis techniques to be used • Ability to negotiate with other divisions to secure resources and adjust a plan • Ability to determine the division of roles and work procedures • Ability to control risks

1-3	Analyze the system and define requirements	• The functions and services required of the system are systematically analyzed and defined. • Network conditions and external interface specifications are analyzed and defined. • Human interface specifications are analyzed and defined. • Restrictive conditions, especially performance, reliability, safety, and maintenance, analyzed and defined.	• Knowledge about the terminology, products, and technology in the users' industry • Knowledge about system analysis techniques (object-oriented analysis techniques, structured analysis techniques, and so on) • Knowledge about writing techniques • Knowledge about requirement systematization techniques • Knowledge about interfaces with peripheral equipment • Knowledge about high-reliability design • Knowledge about hardware restrictions • Knowledge about the system use environment • Knowledge about quality control • Knowledge about risk analysis • Knowledge about control theories and techniques	• Ability to systematically arrange user requirements • Ability to analyze user requirements and define them as system specifications based on the quality secured • Ability to analyze and evaluate system events and to work out system performance and required levels • Ability to reflect restrictions in system requirements • Ability to propose and present system specifications to the users • Ability to negotiate with the users over system specifications (system operation specifications concerning malfunctions as well as normal operations) • Ability to extract risk items and control them, including the measures to deal with them
1-4	Survey related technologies and other companies' intellectual properly rights	• A survey is made of technological trends concerning the system. • A survey is made of intellectual properly rights.	• Knowledge about technological trends in the users' industry • Knowledge about trends in hardware and software configuration methods and element technologies • Knowledge about integrated development environments and development tools • Knowledge about related laws, regulations, and standards, including intellectual properly rights, patents, copyrights, and licenses • Knowledge about survey methods	• Ability to identify the related technologies that should be surveyed • Ability to identify the related laws and regulations that should be surveyed • Ability to carry out surveys quickly
1-5	Study trade-offs between cost, period, and functional volume	• It is determined whether it will be possible to develop the system within a fixed time and budget. • From the above viewpoint, the total balance of system development is examined, and the estimate is reviewed. • Based on the results of the review, a consensus is reached with the users on their requirements.	• Knowledge about estimation techniques • Knowledge about the calculation of the scale of development • Knowledge about development techniques and productivity • Knowledge about cost calculation	• Ability to verify the contents of plans, including the details of work, period, costs, and resources • Ability to coordinate and adjust plans from the viewpoint of the whole system • Ability to control risks • Ability to negotiate with the users over the contents of estimates • Ability to propose and present ideas to the users for negotiation

1-6	Summarize system specifications	• The results of study are sorted out and clarified as system specifications. • Design guidelines for the next process is presented, including design techniques and restrictive conditions. • A study is made how to improve the efficiency of development. • Uncertain factors are sorted out.	• Knowledge about the preparation of system specification documents • Knowledge about design techniques • Knowledge about the reuse of property • Knowledge about writing techniques	• Ability to sort out and write the results of study systematically. • Ability to align the items to be studied from the perspective of the whole system • Ability to determine proper design guidelines based on the results of study • Ability to identify uncertain factors
1-7	Maintain a work plan	• Appropriate measures are studied based on situational changes such as progress and work results, and the study is reflected in the plan.	• Knowledge about budget and performance control • Knowledge about coordination with other divisions • Knowledge about problem solving • Knowledge about project management	• Ability to quickly grasp situational changes • Ability to analyze the causes of situational changes • Ability to coordinate with other divisions about situational changes
1-8	Control work results and retain records	• Records to improve the accuracy of a work plan are retained. • Records to prove the securing of quality by the observance of procedures is retained. • Records to prove the correctness of the references accessed are retained.	• Knowledge about techniques to arrange and analyze data • Knowledge about process control • Knowledge about quality control • Knowledge about intellectual property rights and trade secrets • Knowledge about record control methods	• Ability to quantitatively grasp the project conditions (projected and actual work conditions, quality conditions) • Ability to systematically summarize and analyze information
1-9	Analyze the system and review requirement definitions	• The contents of work are confirmed based on the points to note in the work of system analysis and requirement definition. • The development process and work are confirmed based on the results of the work of system analysis and requirement definition.	• Knowledge about what to be carried out and noted in system analysis and requirement definition • Knowledge about the development process (work procedures) • Knowledge about the industry and products • Knowledge about reviewing techniques	• Ability to point out logically • Ability to present alternatives • Ability to present optimum ideas from total thinking • Ability to evaluate the development process based on actual progress, quality record, and so forth
1-10	Prepare an outline of a project plan	• User requirements, restrictions, work plan, and so forth are summarized as an outline of a project plan. • Appropriate project guidelines are presented. • Measures are presented after the arrangement of risk items.	• Knowledge about the preparation of project plan documents • Knowledge about proposals and planning sheets • Knowledge about project management • Knowledge about writing techniques	• Ability to summarize and write an outline of specifications and restrictions • Ability to align the items to be studied from the perspective of the whole system • Ability to determine the project system • Ability to determine the flow of work and draw up an outline of the plan • Ability to control risks

2. System design

No.	Task	Performance indicators	Required knowledge	Required skill
2-1	Prepare a work plan	• All the work to be performed as system design is extracted. • An efficient work plan is drawn up based on resources. • The division of roles and responsibilities between hardware personnel and software personnel is clarified. • The objects and time of a review and participating members are planned. • The method of extracting and controlling project management items is formulated. • The quality level and evaluation criteria at the end of system design are presented.	• Knowledge about system design techniques and work procedures • Knowledge about co-design • Knowledge about hardware development procedures • Knowledge about software development procedures • Knowledge about quality control items • Knowledge about estimation techniques • Knowledge about personnel education • Knowledge about project planning and control techniques • Knowledge about cost	• Ability to extract work items based on the system design techniques to be used and to systematically arrange them • Ability to prepare concrete measures to secure the required quality • Ability to determine the division of roles and work procedures • Ability to adjust the hardware and the software design schedule through coordination • Ability to work out an appropriate estimate based on the contents of work, period, costs, resources, and so on • Ability to control risks
2-2	Determine the system hardware configuration	• A proper hardware configuration is determined in consideration of the required system quality, including performance, reliability, safety, and maintainability, as well as restrictions.	• Knowledge about the functions, performance, and interfaces of hardware • Knowledge about hardware restrictions • Knowledge about hardware characteristics and specifications • Knowledge about fault tolerance • Knowledge about fail-safe • Knowledge about system LSI	• Ability to select proper hardware from a systematic viewpoint
2-3	Assign system functions to component devices	• The functions required of the system are properly assigned to component devices. • Proper assignment is evaluated.	• Knowledge about the functions, performance, and characteristics of each component device • Knowledge about hardware design • Knowledge about the distribution of functions • Knowledge about the processing of malfunctions and exceptions • Knowledge about multiple control	• Ability to properly assign the required functions based on the functions, performance, and characteristics of individual component devices • Ability to properly assign the processing of malfunctions and exceptions • Ability to properly determine interfaces between component devices

2-4	Determine the division of functions and performance between hardware and software	• Functions are properly divided between hardware and software in consideration of the required system quality, including performance, reliability, safety, and maintainability, as well as restrictions. • The division of functions is reviewed based on the study of the balance of system specifications and user requirements as a whole. • Interfaces between hardware and software are clearly defined.	• Knowledge about the performance, functions, and interfaces of hardware • Knowledge about hardware characteristics • Knowledge about software characteristics • Knowledge about the use of the operating system, language, and commercial library • Knowledge about Java • Knowledge about the method of drawing hardware performance through software • Knowledge about cost	• Ability to identify the proper sharing of the functional volume between hardware and software in accordance with restrictions on the system • Ability to review the balance between the functional volumes of hardware and software from the viewpoint of development efficiency and implementation space • Ability to review the balance between the functional volumes of hardware and software from the viewpoint of hardware characteristics
2-5	Verify feasibility and perform a design review	• Feasibility is verified using development techniques such as prototyping and simulation. • The contents of the design are verified based on the points to note in system design work. • The validity and consistency of the system are verified after the scrutiny of the contents of study.	• Knowledge about concrete implementation methods • Knowledge about prototyping techniques • Knowledge about simulation techniques • Knowledge about design review techniques • Knowledge about the matters to be executed and the points to be noted in system design	• Ability to scrutinize the division of functions between hardware and software and the propriety of interfaces • Ability to verify the propriety of implementation methods in consideration of system requirements • Ability to adjust the matters of study from the viewpoint of the whole system • Ability to point out logically • Ability to present alternatives
2-6	Summarize software specifications	• Software specifications are clarified for each device on which software is to be installed after the contents of study are sorted out. • Design guidelines for the next process is presented, including design techniques and restrictive conditions. • Policies are presented for handling exceptions, malfunctions, and hardware failures. • Uncertain factors are extracted, and it is clarified how to deal with them.	• Knowledge about the matters to be determined as software specifications • Knowledge about the real-time operating system • Knowledge about hardware resources • Knowledge about hardware restrictions • Knowledge about writing techniques • Knowledge about the specification language	• Ability to systematically summarize and write the contents of study • Ability to select proper design guidelines (techniques and environments) based on the contents of study • Ability to clarify the specifications concerning hardware resources and restrictions • Ability to study the handling of exceptions, malfunctions, and hardware failures.

2-7	Determine period, cost, and functional volume	• The development period and cost are estimated in consideration of hardware and software development procedures. • The contents of the estimate are reviewed based on the scrutiny of the balance between hardware and software development.	• Knowledge about the productivity of hardware and software development • Knowledge about the processes of hardware and software development • Knowledge about co-design • Knowledge about resource arrangement • Knowledge about cost	• Ability to calculate the man-hours of development based on development productivity • Ability to adjust hardware and software development schedules • Ability to calculate a proper development period, considering the number of resources and costs
2-8	Maintain a work plan	• Appropriate measures are studied based on situational changes such as progress and work results, and the study is reflected in the plan.	• Knowledge about budget and performance control • Knowledge about coordination with other divisions • Knowledge about problem solving • Knowledge about project management	• Ability to quickly grasp situational changes • Ability to analyze the causes of situational changes • Ability to coordinate with other divisions about situational changes
2-9	Control work results and retain records	• Progress is controlled in such a way as to adjust progress between hardware and software development. • Items necessary for quality control are retained as records. • Records of access to intellectual properly rights are retained.	• Knowledge about techniques to arrange and analyze data • Knowledge about intellectual property rights and trade secrets • Knowledge about record control methods	• Ability to control the project • Ability to systematically summarize and analyze information
2-10	Review the system design	• The contents of the design are verified based on the consistency of the whole system and the points to note in the system design work. • The development process and the contents of the design are verified based on the results of the system design work. • The design quality level is evaluated by the evaluation criteria at the end of system design.	• Knowledge about the matters to be executed and the points to be noted in system design • Knowledge about the development process (work procedures) • Knowledge about the industry and products. • Knowledge about evaluation criteria • Knowledge about reviewing techniques	• Ability to point out logically • Ability to present alternatives • Ability to present optimum ideas from total thinking • Ability to evaluate design quality • Ability to evaluate the development process based on actual progress, quality record, and so forth

3. Development plan

No.	Task	Performance indicators	Required knowledge	Required skill
3-1	Development plan	• The scale of development and the man-hours of development are estimated. • Personnel and education plans are formulated in accordance with estimates. • A development process plan (schedule) is formulated. • Development control policies and systems are formulated. • Estimates and development control systems are reviewed from the viewpoint of the whole system, including hardware.	• Knowledge about process planning procedures • Knowledge about development work procedures • Knowledge about design and development techniques • Knowledge about estimation techniques • Knowledge about personnel education • Knowledge about project planning and control techniques • Knowledge about cost	• Ability to collect, sort out, and analyze estimation-related information, such as specification surveys and past data • Ability to negotiate with other divisions to secure resources • Ability to evaluate the skills of personnel and train them based on development techniques, environments, and so on • Ability to estimate costs • Ability to evaluate cost estimates • Ability to determine a development process and a development control system for each stage of the process • Ability to coordinate the process plan with a hardware development plan • Ability to control risks
3-2	Development environment preparation plan	• Proper development environments and development tools and equipment are selected and arranged in consideration of effects on the process and skills of personnel. • Standardization and reuse are promoted to improve development efficiency and quality.	• Knowledge about development equipment • Knowledge about development environments and tools • Knowledge about purchasing, rental, and leasing • Knowledge about development standards • Knowledge about reuse	• Ability to understand the impact of installing development environments • Ability to arrange to secure development equipment and tools through appropriate channels • Ability to coordinate with the education division for the tools to be introduced
3-3	Review plan	• The objects, viewpoint, and time of a review and participating members are planned. • A plan is prepared concerning the man-hours of each review and is reflected in the process plan. • The period and the man-hours of modifications based on reviews are reflected in the process plan.	• Knowledge about the objectives and effects of reviews • Knowledge about development work procedures • Knowledge about quality control	• Ability to set proper times for review • Ability to select proper review members • Ability to adjust the process plan

3-4	Hardware-software linking test plan	• Proper test tools and equipment are selected for linking tests. • Efficiency is considered in the study of linking test methods. • A plan to secure equipment and materials is prepared to suit the above test environment. • A preparation plan is formulated with the work items required in preparation for linking tests extracted.	• Knowledge about the processes of hardware and software development • Knowledge about the objectives of linking tests and the items to be confirmed • Knowledge about linking test methods, tools, and materials • Knowledge about productivity in the execution of tests	• Ability to present policies concerning the test environment, including the objectives and methods of tests and the criteria for completion • Ability to coordinate the process plan with a hardware development plan • Ability to extract and arrange for necessary equipment and materials from the test plan • Ability to extract and systematically sort out the items to be executed in linking tests
3-5	Embedded-system quality assurance plan	• The quality level (target values) and evaluation criteria at the end of each stage of the development process are set. • Guidelines for the observance of quality targets are presented. • A mechanism to collect and analyze information on quality condition is built. • Those mentioned above are summarized as a quality assurance plan.	• Knowledge about quality items in embedded-system development • Knowledge about the methods of utilizing quality data • Knowledge about the methods of securing quality	• Ability to clarify the required quality • Ability to formulate concrete measures to secure quality • Ability to systematically summarize, analyze, and utilize quality data
3-6	Plan conferences with the hardware group and review the development plan	• The time and objects of coordination with the hardware group are clarified. • The plan is quickly reviewed in response to situational changes, as by reflecting the results of coordination.	• Knowledge about the hardware development process • Knowledge about the items to be coordinated with hardware	• Ability to prepare a plan in consideration of the risk of rework • Ability to quickly respond to the risk of alteration of the plan by always considering measures
3-7	Configuration control plan	• Configuration control guidelines are clearly presented. • A plan is prepared that covers the time to start configuration control, objects, control method, and control system.	• Knowledge about the objectives and methods of configuration control together with the points to note • Knowledge about configuration control tools	• Ability to select appropriate configuration control tools for the system
3-8	Maintenance plan	• A maintenance plan is prepared for each maintenance item.	• Knowledge about the objectives of maintenance work and the time to start maintenance • Knowledge about the industry	• Ability to clarify the scope and period of maintenance work • Ability to extract the items of maintenance work

3-9	Prepare a development plan	• A development plan is prepared by summarizing the items of study. • Proper development guidelines are presented. • Measures are presented based on the arrangement of risk items.	• Knowledge about the items to be clarified in a development plan • Knowledge about the flow of software development • Knowledge about development management • Knowledge about writing techniques	• Ability to systematically summarize and write the items of study • Ability to coordinate the items of study from the viewpoint of the whole system • Ability to clarify the interfaces with other divisions • Ability to coordinate the plan with a hardware development plan • Ability to control risks

4. Software design

No.	Task	Performance indicators	Required knowledge	Required skill
4-1	Prepare a work plan	• All the work to be performed as system design is extracted. • An efficient work plan is drawn up based on resources. • The objects and time of a review and participating members are planned. • The method of extracting and controlling project management items is formulated.	• Knowledge about work procedures in software design • Knowledge about software design and development techniques • Knowledge about quality control items • Knowledge about estimation techniques • Knowledge about personnel education • Knowledge about project planning and control techniques • Knowledge about cost	• Ability to extract and systematically sort out work items in line with development techniques • Ability to formulate concrete measures to secure the required quality • Ability to determine the division of roles and work procedures • Ability to calculate reasonable estimates based on the contents of work, period, cost, resources, and so on • Ability to control risks
4-2	Determine the software structure	• Guidelines for designing an optimum task configuration are presented by effectively utilizing the operating system functions. • Guidelines for designing common data are presented based on the consistency of the system. • Operating system resources and classes are defined, and design guidelines are presented. • Modular configuration design guidelines are presented based on the consideration of module partition, standardization, reuse, feasibility, and so forth.	• Knowledge about software design techniques • Knowledge about task configuration, common data, operating system resources, and class design concepts • Knowledge about object-oriented design and structured design • Knowledge about software characteristics • Knowledge about standardization • Knowledge about the reuse of property	• Ability to clarify software structure design guidelines based on system requirement specifications • Ability to realize a proper structure in terms of expandability, flexibility, reliability, and so forth • Ability to utilize real-time operating system functions • Ability to the method of drawing hardware performance through software
4-3	Perform a design review	• The contents of design are verified based on the points to note in software design work. • The validity and consistency of the system are verified after the scrutiny of the contents of study.	• Knowledge about software design techniques • Knowledge about the matters to be executed and the points to be noted in software design • Knowledge about design review techniques	• Ability to align the items to be studied from the perspective of the whole system • Ability to point out logically • Ability to present alternatives

4-4	Summarize software design	• The software design is summarized after it is confirmed that there is no inconsistency as a result of overall study of software components, including task configuration and common data. • Uncertain factors are extracted, and it is clarified how to deal with them. • Design guidelines for the next process is presented.	• Knowledge about the matters to be studied in software design • Knowledge about writing techniques	• Ability to sort out the matters to be studied and summarize them as software specifications • Ability to select appropriate design techniques for the system • Ability to select an appropriate development environment for the system
4-5	Maintain a work plan	• Appropriate measures are studied based on situational changes such as progress, quality condition, and work results, and the study is reflected in the plan.	• Knowledge about budget and performance control • Knowledge about coordination with other divisions • Knowledge about problem solving • Knowledge about project management	• Ability to quickly grasp situational changes • Ability to analyze the causes of situational changes • Ability to coordinate with other divisions about situational changes
4-6	Hold a conference for coordination with hardware design	• The time and objects of coordination with the hardware group are clarified. • The plan is quickly reviewed in response to situational changes, as by reflecting the results of coordination.	• Knowledge about the hardware development process • Knowledge about the items to be coordinated with hardware	• Ability to prepare a plan in consideration of the risk of rework • Ability to quickly respond to the risk of alteration of the plan by always considering measures
4-7	Control work results and retain records	• Progress is quantitatively grasped and utilized for control. • The items necessary for development management are retained as records.	• Knowledge about the data subject to development management • Knowledge about data arrangement and analysis techniques • Knowledge about record control methods	• Ability to manage the project • Ability to systematically sort out and analyze information
4-8	Review the software design	• The contents of the design are verified based on the consistency of the whole system and the points to note in the software design work. • The development process and the contents of the design are verified based on the results of the software design work. • The design quality level is evaluated by the evaluation criteria at the end of system design.	• Knowledge about the matters to be executed and the points to be noted in software design • Knowledge about the development process (work procedures) • Knowledge about the industry and products. • Knowledge about evaluation criteria • Knowledge about reviewing techniques	• Ability to point out logically • Ability to present alternatives • Ability to present optimum ideas from total thinking • Ability to evaluate design quality • Ability to evaluate the development process based on actual progress, quality record, and so forth

5. Create and test programs

No.	Task	Performance indicators	Required knowledge	Required skill
5-1	Prepare a work plan	• All the work to be performed from programming to program tests is extracted. • An efficient work plan is drawn up based on resources. • The objects and time of a review and participating members are planned. • The method of extracting and controlling project management items is formulated.	• Knowledge about work procedures from programming to program tests • Knowledge about programming and program test techniques • Knowledge about quality control items • Knowledge about estimation techniques • Knowledge about personnel education • Knowledge about project planning and control techniques • Knowledge about cost	• Ability to extract work items based on programming and program test techniques and to systematically arrange them • Ability to prepare concrete measures to secure the required quality • Ability to determine the division of roles and work procedures • Ability to work out an appropriate estimate based on the contents of work, period, costs, resources, and so on • Ability to control risks
5-2	Create programs and extract program test items	• Programming guidelines are presented, covering standardization, the reuse of property, and so on. • A program creation environment is built. • The scope of program tests (those conducted on software alone) and that of tests conducted on software linked to hardware are clearly defined. • Program test guidelines are presented. • The scope to be confirmed in program tests is clarified, and all test items are extracted.	• Knowledge about programming techniques • Knowledge about programming efficiency • Knowledge about programming quality • Knowledge about program test techniques • Knowledge about program test procedures • Knowledge about program test environments • Knowledge about the point of view of program tests	• Ability to clarify program creation guidelines based on software specifications • Ability to introduce measures to improve program creation efficiency and secure quality • Ability to extract program test items and systematically sort them out based on the point of view of program tests • Ability to determine an program test environment • Ability to determine the sequence of program tests

5-3	Perform a code review and a design review of program test items	• The point of view of a code review is presented. • A target number of bugs to be extracted in a code review is set. • The consistency between software specifications and the contents of programming, the procedures to access hardware and operating system resources, and the correctness of interrupt control and so on are evaluated. • The correctness and validity of test items are evaluated based on program test guidelines. • It is ascertained that the requirements of each program (functions, performance) can be confirmed by program tests.	• Knowledge about hardware specifications • Knowledge about operating system resources and interrupt control • Knowledge about processing efficiency • Knowledge about structured design techniques and modular design techniques • Knowledge about program standardization • Knowledge about test techniques • Knowledge about the point of view of tests	• Ability to align the contents of individual modules' processing based on the operations of programs as a whole • Ability to verify the consistency between the requirements of each module and the methods of realizing them • Ability to evaluate hardware access methods • Ability to point out logically • Ability to present alternatives
5-4	Summarize program quality data	• Program quality data is defined. • Program quality data is collected and systematically summarized. • The summarized quality data is analyzed and utilized for quality evaluation.	• Knowledge about the meanings and the objectives of use of program quality data • Knowledge about interrelations among the groups of quality data • Knowledge about the causes of bugs • Knowledge about writing techniques	• Ability to analyze and evaluate the collected quality data • Ability to propose measures for improving quality based on the results of analysis
5-5	Perform program tests	• A program test environment is prepared. • Guidelines concerning test efficiency are presented. • The causes of the occurrence of bugs are identified. • The scope of confirmed effects of bug correction is clarified. • The test execution plan is reviewed after quality condition is evaluated and predicted in accordance with the condition of occurrence of bugs.	• Knowledge about the point of view of confirmation in program tests • Knowledge about test techniques • Knowledge about test procedures • Knowledge about test environments • Knowledge about the handling of occurrences of bugs • Knowledge about quality evaluation and prediction techniques	• Ability to guide the method of conducting program tests • Ability to analyze the causes of occurrences of bugs • Ability to collect and analyze the conditions of occurrence of bugs • Ability to review test items based on the conditions of occurrence of bugs and the contents of bugs • Ability to control the conditions of bugs that have occurred • Ability to coordinate with other divisions
5-6	Summarize program test results	• Quality data is defined. • Quality data is collected and systematically summarized. • Quality data is analyzed and utilized for quality evaluation.	• Knowledge about the analysis of the conditions of occurrence of bugs • Knowledge about quality evaluation • Knowledge about the methods of utilizing quality data • Knowledge about writing techniques	• Ability to evaluate quality by analyzing the contents of bugs and the conditions of occurrence • Ability to analyze quality condition

5-7	Maintain the work plan	• Appropriate measures are studied based on situational changes such as progress, quality condition, and work results, and the study is reflected in the plan.	• Knowledge about budget and performance control • Knowledge about coordination with other divisions • Knowledge about problem solving • Knowledge about project management	• Ability to quickly grasp situational changes • Ability to analyze the causes of situational changes • Ability to coordinate with other divisions about situational changes
5-8	Hold a conference for coordination with hardware design	• The time and objects of coordination with the hardware group are clarified. • The plan is quickly reviewed in response to situational changes, as by reflecting the results of coordination.	• Knowledge about the hardware development process • Knowledge about the items to be coordinated with hardware	• Ability to prepare a plan in consideration of the risk of rework • Ability to quickly respond to the risk of alteration of the plan by always considering measures
5-9	Control work results and retain records	• Progress is quantitatively grasped and utilized for control. • The items necessary for development management are retained as records. • Software version management is performed.	• Knowledge about the data subject to development management • Knowledge about data arrangement and analysis techniques • Knowledge about record control methods • Knowledge about version management	• Ability to manage the project • Ability to systematically sort out and analyze information • Ability to understand software components and control their history
5-10	Review programming and program tests	• The contents of the design and tests are verified based on the points to note in programming and program tests. • The development process, test results, and the contents of modifications are verified based on the results of programming and program tests. • The quality level is evaluated by the evaluation criteria at the end of program tests.	• Knowledge about the matters to be executed and the points to be noted in programming and program tests • Knowledge about the development process • Knowledge about evaluation criteria • Knowledge about reviewing techniques	• Ability to point out logically • Ability to present alternatives • Ability to present optimum ideas from total thinking • Ability to evaluate program quality • Ability to evaluate the development process based on actual progress, quality record, and so forth

6. Hardware-software linking tests

No.	Task	Performance indicators	Required knowledge	Required skill
6-1	Prepare a work plan	• All the work to be performed in linking tests is extracted. • An efficient work plan is drawn up based on resources. • The objects and time of a review and participating members are planned. • The method of extracting and controlling project management items is formulated.	• Knowledge about work procedures and techniques in linking tests • Knowledge about quality control items • Knowledge about estimation techniques • Knowledge about personnel education • Knowledge about project planning and control techniques • Knowledge required to perform effective reviews • Knowledge about cost	• Ability to extract and systematically sort out work items in line with linking test techniques • Ability to formulate concrete measures to secure the required quality • Ability to determine the division of roles and work procedures • Ability to coordinate the pace of work with the hardware group • Ability to calculate reasonable estimates based on the contents of work, period, cost, resources, and so on • Ability to control risks
6-2	Build a test environment	• Test environment guidelines are presented. • All necessary work to prepare for tests is extracted. • Test equipment and tools to be used in tests are properly selected and arranged for. • A test environment is built, including the connection of equipment and tools, based on the size of test personnel and the flow of tests.	• Knowledge about linking test equipment • Knowledge about linking test environments and tools • Knowledge about purchasing, rental, and leasing • Knowledge required to connect linking test equipment • Knowledge about the linking test process	• Ability to arrange to secure test equipment and tools through appropriate channels • Ability to coordinate with the education division about test tools • Ability to build a test environment by connecting test equipment

6-3	Extract test items, determine test procedures, and review them	• Linking test guidelines are presented. • The scope of program tests (those conducted on software alone) and that of tests conducted on software linked to hardware are clearly defined. • All test items are extracted after the scope to be confirmed by linking tests is clarified. • Efficient work procedures are drawn up based on resources. • The correctness and validity of test items are evaluated based on linking test guidelines. • Test items are evaluated from the viewpoint of confirming user requirements.	• Knowledge about the objectives and techniques of linking tests • Knowledge about linking test procedures • Knowledge about the equipment used for linking tests • Knowledge about the point of view of linking tests • Knowledge about the reuse of property	• Ability to extract and systematically sort out work items from the point of view of linking tests • Ability to determine the sequence of linking tests in consideration of the number of devices to be used • Ability to ascertain that system requirements (functions, performance) can be confirmed by linking tests • Ability to point out logically • Ability to present alternatives
6-4	Conduct tests	• Guidelines for efficient testing are presented. • The causes of the occurrence of bugs are identified. • The scope of confirmed effects of bug correction is clarified. • The test execution plan is reviewed after quality condition is evaluated and predicted in accordance with the condition of occurrence of bugs. • Measures are taken in accordance with test results in coordination with the hardware group. • Quality data is defined. • Quality data is collected and systematically summarized. • Quality data is analyzed and utilized for quality evaluation.	• Knowledge about the point of view of confirmation in linking tests • Knowledge about test equipment • Knowledge about test techniques • Knowledge about test procedures • Knowledge about test environments • Knowledge about the handling of occurrences of bugs • Knowledge about quality evaluation and prediction techniques • Knowledge about the analysis of the conditions of occurrence of bugs • Knowledge about the methods of utilizing quality data • Knowledge about writing techniques	• Ability to guide test methods, including the method of using equipment • Ability to analyze the causes of occurrences of bugs (ability to isolate hardware and software) • Ability to collect and analyze the conditions of occurrence of bugs • Ability to review test items based on the conditions of occurrence of bugs and the contents of bugs • Ability to control the conditions of bugs that have occurred • Ability to work in cooperation with the hardware group • Ability to coordinate with other divisions • Ability to evaluate quality by analyzing the contents of bugs and the conditions of occurrence • Ability to analyze quality condition
6-5	Maintain a work plan	• Appropriate measures are studied based on situational changes such as progress, quality condition, and work results, and the study is reflected in the plan.	• Knowledge about budget and performance control • Knowledge about coordination with other divisions • Knowledge about problem solving • Knowledge about project management	• Ability to quickly grasp situational changes • Ability to analyze the causes of situational changes • Ability to coordinate with other divisions about situational changes

6-6	Align and coordinate with the hardware group	• The time and objects of coordination with the hardware group are clarified. • The plan is quickly reviewed in response to situational changes, as by reflecting the results of coordination.	• Knowledge about the hardware development process • Knowledge about the items to be coordinated with hardware	• Ability to prepare a plan in consideration of the risk of rework • Ability to quickly respond to the risk of alteration of the plan by always considering measures
6-7	Control work results and retain records	• Progress is quantitatively grasped and utilized for control. • The items necessary for development management are retained as records. • Software version management is performed.	• Knowledge about the data subject to development management • Knowledge about data arrangement and analysis techniques • Knowledge about record control methods • Knowledge about version management	• Ability to manage the project • Ability to systematically sort out and analyze information • Ability to understand software components and control their history
6-8	Review linking tests	• The contents of tests are verified based on the points to note in linking tests. • The development process, test results, and the contents of modifications are verified based on the results of linking tests. • The quality level is evaluated by the evaluation criteria at the end of linking tests.	• Knowledge about the matters to be executed and the points to be noted in linking tests • Knowledge about the development process • Knowledge about evaluation criteria • Knowledge about reviewing techniques	• Ability to point out logically • Ability to present alternatives • Ability to present optimum ideas from total thinking • Ability to evaluate system quality • Ability to evaluate the development process based on actual progress, quality record, and so forth

7. System evaluation

No.	Task	Performance indicators	Required knowledge	Required skill
7-1	Prepare a plan to receive an examination	• Guidelines are presented for the arrangement of data on the products and the results of control. • All items of work to be performed are extracted for the arrangement of data on the products and the results of control required to receive inspection. • An effective work plan is drawn up based on resources.	• Knowledge about the objectives of receiving inspection • Knowledge about inspection receiving procedures • Knowledge about the products and the results of control required to receive inspection • Knowledge about estimation techniques	• Ability to extract and systematically arrange work items in accordance with the guidelines for arranging the data on the products and the results of control required to receive inspection • Ability to determine the division of roles and work procedures • Ability to calculate reasonable estimates based on the contents of work, period, cost, resources, and so on • Ability to coordinate with other divisions about the time of receiving inspection and so on
7-2	Prepare to receive inspection	• The data on the products and the results of control is arranged in accordance with the quality level required in receiving inspection and the evaluation criteria.	• Knowledge about the objectives of receiving inspection • Knowledge about the quality level required in receiving inspection and the evaluation criteria • Knowledge about the items and contents to be written as data on the products and the results of control • Knowledge about writing techniques	• Ability to evaluate the quality level of data on the products and the results of control • Ability to formulate the measures to grasp progress in work
7-3	Evaluate the system	• The system is evaluated in line with the system and items of system evaluation. • If there are any items not meeting evaluation criteria, the causes are analyzed and concrete improvements are presented. • The results of system evaluation are accumulated as assets.	• Knowledge about the objectives of system evaluation • Knowledge about the system and items of system evaluation • Knowledge about system evaluation criteria and evaluation methods • Knowledge about problem analysis and solving techniques • Knowledge about the system industry and products • Knowledge about quality control and evaluation • Knowledge about record control methods	• Ability to evaluate the consistency with user requirements • Ability to manage evaluation results as records
7-4	Transfer the system	• Data required for system transfer is prepared. • Transfer work is being carried out in coordination with the destination.	• Knowledge about transfer procedures • Knowledge about the data required at the time of transfer • Knowledge about writing techniques	• Ability to guide the preparation of the data required at the time of transfer • Ability to coordinate with other divisions

8. Evaluate the project

No.	Task	Performance indicators	Required knowledge	Required skill
8-1	Evaluate the project	• The project is evaluated in line with the project evaluation system. • If there are any items not meeting evaluation criteria, the causes are analyzed and concrete improvements are presented. • The results of project evaluation are accumulated as assets.	• Knowledge about the objectives of system evaluation • Knowledge about the system and items of system evaluation • Knowledge about system evaluation criteria and evaluation methods • Knowledge about problem analysis and solving techniques • Knowledge about the system industry and products • Knowledge about quality control and evaluation • Knowledge about cost • Knowledge about patents • Knowledge about development work procedures • Knowledge about personnel education • Knowledge about project planning and management techniques • Knowledge about record control methods	• Ability to evaluate the project against targets • Ability to manage evaluation results (know-how) as records

9. Maintain software

No.	Task	Performance indicators	Required knowledge	Required skill
9-1	Prepare a maintenance plan	• The definition of maintenance work is clarified. • All concrete items to be performed are extracted after the clarification of corresponding work. • An effective maintenance work plan is drawn up based on resources. • The method of extracting and controlling maintenance control items is formulated.	• Knowledge about the objectives of maintenance • Knowledge about the system industry • Knowledge about the kinds of maintenance work • Knowledge about handling methods suitable for objectives of maintenance • Knowledge about the data required in performing maintenance work	• Ability to extract and systematically arrange concrete work items in line with the definition of maintenance work (corresponding work) • Ability to clarify the scope of effects of performing maintenance work • Ability to determine the division of roles, work procedures, and maintenance system • Ability to calculate reasonable estimates based on the contents of work, period, cost, resources, and so on • Ability to clarify maintenance work procedures
9-2	Arrange development information for maintenance	• The information required for maintenance is controlled. • Alteration control of information for maintenance is performed.	• Knowledge about document control methods and procedures • Knowledge about alteration control	• Ability to define the information required in maintenance • Ability to systematically arrange and control information for maintenance • Ability to clarify the procedures for altering information for maintenance
9-3	Prepare a maintenance environment	• A maintenance environment for version management is prepared. • A maintenance environment for user information management is prepared. • A development environment is built at the time of occurrence of maintenance.	• Knowledge about version management methods • Knowledge about user information management methods • Knowledge about the maintenance of development environments	• Ability to study and define version management procedures • Ability to study and define user management procedures • Ability to study the method of maintaining a development environment and restore it at the time of occurrence of maintenance
9-4	Carry out maintenance	• The objectives of maintenance are clarified. • A maintenance system is arranged promptly for the performance of work. • The results of performance of maintenance are accumulated as assets.	• Knowledge about concrete work items corresponding to each kind of maintenance work • Knowledge about update procedures • Knowledge about writing techniques	• Ability to build a system suitable for the need of maintenance and to take proper measures • Ability to coordinate with other divisions • Ability to study and take emergency measures • Ability to identify the scope of effects of performing maintenance and to take proper measures

9-5	Transfer the system	• Data required for system transfer is prepared. • Transfer work is being carried out in coordination with the destination.	• Knowledge about transfer procedures • Knowledge about the data required at the time of transfer • Knowledge about writing techniques • Knowledge about cost price	• Ability to guide the preparation of the data required at the time of transfer • Ability to coordinate with other divisions • Ability to adjust the time of transfer

4. Body of Knowledge

In the body of knowledge for technical engineers (embedded systems), the knowledge which is needed to successfully perform the activities described in 2, "Key Activities," and to solve various problems is divided into groups according to technical and problem-solving concepts and is classified in a hierarchical structure. Here, the various problems may include quality degradation, cost increase, and development delay.

The body of knowledge which technical engineers (embedded systems) must have consists of the following two kinds:

1) IT common body of knowledge
2) Practical and core bodies of knowledge for technical engineers (embedded systems)

The "IT common body of knowledge" in 1) is not limited to technical engineers (embedded systems), but it is necessary for examinees of all examination categories. It is therefore provided in a separate volume. For details, refer to the "Inform Technology Engineers Skill Standards: IT Body of Knowledge."

By consulting "Information Technology Engineers Examinations: Scope of Examinations," we can know that technical engineers (embedded systems) are tested for knowledge at the following technical levels in the five fields of the IT common body of knowledge:

"II. Computer systems (level III)"
"III. System development and operation (level II)"
"IV. Network technology (level II)"
"VI. Security (level II)"
"VII. Standardization (level II)"

In 2), "Practical and core bodies of knowledge for technical engineers (embedded systems)," the part corresponding to the practical body of knowledge is classified into A, "Embedded-system development process," the field where technical engineers (embedded systems) should show their competence most, and G, "New application fields of embedded systems." The part corresponding to the core body of knowledge is classified into B, "Software technology for embedded systems," C, "Hardware technology for embedded systems," E, "Control theories and technologies for embedded systems," D, "Embedded-system development environments," and F, "Use of object-oriented technology." B, C, and E are required for logically building systems that are effective and refined both technologically and economically. D and F provide a base for efficient development of embedded systems.

(Note) The following factors described in the "Practical and core bodies of knowledge for technical engineers (embedded systems)" are trademarks or product names of individual manufacturers:

- Java2 MicroEdition, Java TV API, Java Auto API
- Jini, Embedded Java, HotSpot
- Memory Stick, Smart Media, Multimedia Card, SD Card

[Practical and core bodies of knowledge for technical engineers (embedded systems)]

Knowledge field	Major classification	Intermediate classification	Minor classification
A. Embedded-system development process	1. System analysis and requirement definition	1.1 Acquiring and adjusting requirements	1.1.1 Understanding the field in question 1.1.2 Hearing from the customers, followed by confirmation and adjustment 1.1.3 Market research 1.1.4 Related laws, regulations, and standards
		1.2 System analysis and requirement definition	1.2.1 Systematic analysis and definition of functional requirements 1.2.2 Analysis and definition of network conditions and external interface specifications 1.2.3 Analysis and definition of human interface specifications 1.2.4 Analysis and definition of performance conditions 1.2.5 Analysis and definition of reliability and safety 1.2.6 Analysis and definition of maintenance conditions 1.2.7 Analysis and definition of other restrictive conditions (such as hardware) 1.2.8 Feasibility analysis 1.2.9 Risk analysis 1.2.10 Review of the contents of requirement definitions
		1.3 Related technologies and other companies' intellectual property rights	1.3.1 General technologies in a new field 1.3.2 Trends in hardware/software configurations and element technologies 1.3.3 Trends in development technology, environment, and tools 1.3.4 Contents of other companies' technologies 1.3.5 Methods of dealing with other companies' technologies 1.3.6 Product liability
		1.4 Trade-offs between cost, period, and functional volume	1.4.1 Estimating work volume and period 1.4.2 Estimating resource volume and cost 1.4.3 Scrutiny of required items 1.4.4 Collecting and analyzing a track record of similar projects 1.4.5 Coordinating and negotiating with the customers
		1.5 System specifications	1.5.1 System specifications 1.5.2 Formulation of design policies 1.5.3 Reuse of existing property 1.5.4 Handling uncertain factors
		1.6 Project plan	

	2. System design		
		2.1 Hardware configuration of the system	
			2.1.1 Classifying the system into component devices 2.1.2 Devices to share the functions of interfacing external equipment 2.1.3 Communication method between devices 2.1.4 Internal configurations of devices 2.1.5 Inter-unit interfacing methods
		2.2 Assignment of system functions to component devices	
			2.2.1 Adding and evaluating functions for divided arrangement of required functions 2.2.2 Adding and evaluating initialization functions 2.2.3 Adding and evaluating RAS (Reliability, Availability and Serviceability) functions 2.2.4 Adding and evaluating multiplexing control functions (processors and communication paths) 2.2.5 Evaluating the interfacing methods between devices and between units 2.2.6 Defining the interfaces between devices and between units
		2.3 Division of functions and performance between hardware and software	
			2.3.1 Division of functions between hardware and software 2.3.2 Defining hardware-software interface specifications 2.3.3 Division of the means of achieving system performance requirements 2.3.4 Selecting microprocessors to realize hardware function specifications and evaluating the making of system LSI
		2.4 Feasibility verification and design review	
			2.4.1 Feasibility verification by experiment 2.4.2 Design review
		2.5 Software specifications	
			2.5.1 Software requirement specifications 2.5.2 Items to be described in software specifications 2.5.3 Software design conditions 2.5.4 Handling uncertain factors
		2.6 Determining period, cost, and functional volume	
			2.6.1 Estimating work items, work volume, period, development resources, and cost 2.6.2 Calculating development period, cost, and functional volume
	3. Development plan		
		3.1 Process plan	
			3.1.1 Preparing a process plan 3.1.2 Preparing an estimate 3.1.3 Preparing a process control plan 3.1.4 Preparing an education plan
		3.2 Development environment preparation plan	
			3.2.1 Making a plan to prepare a development tool environment 3.2.2 Making a plan to prepare development equipment 3.2.3 Preparing a standardization plan

		3.3 Review plan

		3.4 Hardware-software linking test plan	
			3.4.1 Setting the time to start linking tests 3.4.2 Preparing a test location plan 3.4.3 Making a test preparation plan 3.4.4 Preparing a personnel plan
		3.5 Embedded-system quality assurance plan	
			3.5.1 Preparing a quality target plan 3.5.2 Preparing a quality control plan
		3.6 Planing conferences with the hardware group and the development plan	
			3.6.1 Preparing plans for conferences with the hardware group 3.6.2 Reviewing the development plan
		3.7 Configuration control plan	
			3.7.1 Documents subject to configuration control 3.7.2 Configuration control operation rules
		3.8 Maintenance plan	
			3.8.1 Defining maintenance 3.8.2 Maintenance conditions 3.8.3 Preparing a maintenance plan
		3.9 Development plan	
	4. Software design		
		4.1 Software structure	
			4.1.1 Task structure design 4.1.2 Thread 4.1.3 Design of common data 4.1.4 Defining operating system resources 4.1.5 Class design 4.1.6 Module structure design 4.1.7 Memory layout
		4.2 Design review	
			4.2.1 Method of performing work and confirmation of its contents 4.2.2 Confirming the realization of functions (services) 4.2.3 Evaluating the task structure design 4.2.4 Evaluating the task design 4.2.5 Evaluating the module structure design
		4.3 Software design	

- 5. Program creation and program tests
  - 5.1 Program creation and program test items
    - 5.1.1 Standard format for describing source programs
    - 5.1.2 Programming environment
    - 5.1.3 Creating programs
    - 5.1.4 Test methods
    - 5.1.5 Program unit-test items
    - 5.1.6 Program test items
    - 5.1.7 Fault isolation procedures
  - 5.2 Code review and a design review of program test items
    - 5.2.1 Code review
    - 5.2.2 Reviewing test methods and test items
    - 5.2.3 Review check list
  - 5.3 Program quality data
    - 5.3.1 Program quality data items
    - 5.3.2 Evaluating program quality data
  - 5.4 Program tests
    - 5.4.1 Building and maintaining a program unit-test environment
    - 5.4.2 Performing program unit tests
    - 5.4.3 Editing the system
    - 5.4.4 Building and maintaining a program test environment
    - 5.4.5 Performing program tests
- 6. Hardware-software linking tests
  - 6.1 Test environment
    - 6.1.1 Test equipment and materials
    - 6.1.2 Place to conduct tests
    - 6.1.3 Test personnel
  - 6.2 Extracting test items, test procedures, and review
    - 6.2.1 Contents of tests
    - 6.2.2 Linking test strategy
    - 6.2.3 Order of conducting tests
    - 6.2.4 Means of testing
    - 6.2.5 Extracting test items
    - 6.2.6 Setting test evaluation criteria
    - 6.2.7 Test plan
    - 6.2.8 Reviewing linking test specifications
  - 6.3 Conducting tests
    - 6.3.1 Building and maintaining a test environment
    - 6.3.2 Conducting tests
    - 6.3.3 Negotiating with other divisions
    - 6.3.4 Dealing with contingencies
    - 6.3.5 Isolating faults

	7. System evaluation		
		7.1 Receiving inspection	
			7.1.1 Preparing a plan to receive inspection
			7.1.2 Objects of inspection
		7.2 Evaluating the system	
			7.2.1 Analyzing inspection results and taking measures
			7.2.2 System evaluation items and criteria
			7.2.3 System evaluation procedures
		7.3 Transferring the system	
	8. Evaluating the project		
		8.1 Evaluating the project	
			8.1.1 Project evaluation items and criteria
			8.1.2 Project evaluation procedures
	9. Maintaining software		
		9.1 Preparing a maintenance plan	
		9.2 Arranging information for maintenance	
			9.2.1 Specification sheets
			9.2.2 Design sheets
			9.2.3 Programs
			9.2.4 Development records
			9.2.5 Data on the method of operating the development environment
			9.2.6 Record of revisions
		9.3 Preparing a maintenance environment	
			9.3.1 Version number control system
			9.3.2 Customer information control system
			9.3.3 Maintenance environment
		9.4 Carrying out maintenance	
			9.4.1 Types of maintenance (fault correction and functional improvement)
			9.4.2 Control items (alteration control and complaint control)
			9.4.3 Reflecting in the next project
		9.5 Transferring the system	
			9.5.1 Objects of transfer
			9.5.2 Transfer procedures

	10. Common to Development Processes		
		10.1 Preparing a work plan	
			10.1.1 Preparing and concretizing a work plan 10.1.2 Progress and quality control methods 10.1.3 Items to be controlled 10.1.4 Identifying work items and assigning workers 10.1.5 Preparing a design review execution plan 10.1.6 Preparing the assignment of priorities of using the environment
		10.2 Maintaining a work plan	
			10.2.1 Performing work control 10.2.2 Performing progress and quality control 10.2.3 Progress in hardware development 10.2.4 Executing a detailed work plan
		10.3 Work result control and records	
			10.3.1 Performing progress control and recording progress results 10.3.2 Performing quality control and recording quality 10.3.3 Recording access to other companies' data 10.3.4 Product structure and version number change 10.3.5 History of program alterations 10.3.6 Recording root causes of bugs and methods of dealing with them
		10.4 Aligning and coordinating with the hardware group	
			10.4.1 Holding a coordination conference with the hardware group 10.4.2 Preparing a rework plan 10.4.3 Executing rework
		10.5 Reviewing system analysis and requirement definition	
			10.5.1 Planing a review 10.5.2 Holding a review 10.5.3 Dealing with items pointed out in a review 10.5.4 Judge whether the process reviewed has been completed

Knowledge field	Major classification	Intermediate classification	Minor classification
B. Software technology for embedded systems			
	1. Basic principles of embedded system software		
		1.1 Real-time processing	
			1.1.1 Time-limited processing
			1.1.2 Multitask processing
			1.1.3 Processing priorities
			1.1.4 Thread
		1.2 Event-driven	
			1.2.1 Scheduling method
			1.2.2 Interruption and event-driven
			1.2.3 Preemptive processing
		1.3 Interruption	
			1.3.1 Interrupt processing and tasks
			1.3.2 Method of notifying occurrence of events
			1.3.3 Effective use of input/output latency
		1.4 Exclusive access control	
			1.4.1 Means of exclusive access control
			1.4.2 Minimizing exclusive access control time
		1.5 Multi-interruption priorities	
			1.5.1 Multi-interruption
			1.5.2 Setting priorities by hardware
			1.5.3 Setting priorities by software
		1.6 Real-time resources	
			1.6.1 MPU resources
			1.6.2 Memory resources
			1.6.3 Program resources
			1.6.4 Input/output resources

	2. Real-time operating system		
		2.1 Task management	
			2.1.1 Task life cycle management
			2.1.2 Context management
			2.1.3 Task scheduling
			2.1.4 Intertask communication
		2.2 Resource allocation and management	
			2.2.1 MPU resource management
			2.2.2 Memory resource management
			2.2.3 Input/output resource management
			2.2.4 Shared resources and exclusive access control
			2.2.5 Synchronous control
		2.3 Interruption processing	
			2.3.1 Significance of interruption
			2.3.2 Multi-interruption priorities
		2.4 System calls	
		2.5 Input/output drivers	
			2.5.1 Characteristics of input/output resources and effective methods of use
			2.5.2 Structure of input/output resource management programs
			2.5.3 File management and device driver call
		2.6 Other real-time operating system functions	
			2.6.1 input/output support function
			2.6.2 System management function
			2.6.3 Object name management function
	3. Real-time applications design and implementation		
		3.1 Task design	
			3.1.1 Task division techniques
			3.1.2 Task relation chart
			3.1.3 Task design techniques
		3.2 Design of program units	
			3.2.1 Loading method
			3.2.2 Loading method and programming unit
			3.2.3 Method of corresponding tasks and programs
		3.3 Sharing routine data	
			3.3.1 Reference method
			3.3.2 Exclusive access control
		3.4 Sharing files	
			3.4.1 Management method
			3.4.2 Exclusive access control
		3.5 Implementation support	
			3.5.1 Tuning technology
			3.5.2 Trade-off
			3.5.3 High reliability
			3.5.4 Selecting a development environment
			3.5.5 Libraries

	4. Real-time kernel design technology		
		4.1 Context design	
			4.1.1 Context design techniques
			4.1.2 Context switching in interruption switching
			4.1.3 Multi-interruption scheduling
			4.1.4 Interruption processing service
			4.1.5 System call
			4.1.6 Task scheduler
		4.2 Exclusive access control	
		4.3 Data structure and processing design	
			4.3.1 Buffer control method and processing method
			4.3.2 Context switching and return in interruption processing
		4.4 Selecting a development environment	
		4.5 Driver design as an application	
			4.5.1 Functions of the kernel
			4.5.2 Processing methods
			4.5.3 Development environment
		4.6 Driver design as operating system functions	
			4.6.1 Functions the kernel must provide
			4.6.2 Processing methods
			4.6.3 Development environment
		4.7 File management design	
			4.7.1 Outline of file management functions
			4.7.2 File allocation method
			4.7.3 Implementation techniques
		4.8 Loader design	
			4.8.1 Loader's role
			4.8.2 Development system
			4.8.3 Implementation techniques
		4.9 Design of initialization functions	
			4.9.1 System development
			4.9.2 System resetting
			4.9.3 Bootstrap loading
			4.9.4 Kernel initialization
			4.9.5 Starting applications
	5. Device driver		
		5.1 Device driver	
			5.1.1 Device driver implementation
			5.1.2 Realization method as an application
			5.1.3 Realization method as a system call function
			5.1.4 Method of realizing a device driver causing no interruption
			5.1.5 Characteristics of each implementation method
			5.1.6 Method of monitoring an interrupt limitation time
			5.1.7 Method of guaranteeing the simultaneous parallelism of device drivers
			5.1.8 Real-time performance
			5.1.9 External interfaces and implementation

Knowledge field	Major classification	Intermediate classification	Minor classification
C. Hardware technology for embedded systems			
	1. Basic architecture of embedded systems		
		1.1 Architecture of embedded systems	
			1.1.1 MPU architecture
			1.1.2 Instruction execution control circuit
		1.2 Types of MPU	
			1.2.1 Instruction control architecture
			1.2.2 Instruction set architecture
			1.2.3 CISC
			1.2.4 RISC
			1.2.5 Architecture and applications of DSP
			1.2.6 Architecture and applications of media processors (for image and voice processing)
			1.2.7 Architecture and applications of ASP (instruction set optimization processor)
		1.3 Multiprocessor system	
			1.3.1 Characteristics of the multiprocessor system
			1.3.2 Multiprocessor system building technology
		1.4 Bus architecture	
			1.4.1 Bus configuration inside an application system
			1.4.2 Standard bus
			1.4.3 Characteristics of buses and considerations
		1.5 Memory types and architectures	
			1.5.1 Memory types
			1.5.2 Memory architectures and characteristics
		1.6 Additional functions	
			1.6.1 Low power consumption mode
			1.6.2 Noise cancellation function
		1.7 Making system LSI	
			1.7.1 ASIC
			1.7.2 FPGA
	2. Architecture around MPU		
		2.1 Interruption technology	
			2.1.1 Interruption factors
			2.1.2 Method of increasing the number of interrupts
			2.1.3 Timing of accepting interrupts and response time up to interruption processing
		2.2 DMA technology	
			2.2.1 Transfer mode
			2.2.2 Relations between DMA and cache
			2.2.3 DMA and virtual memory

		2.3 Cache memory technology	
			2.3.1 Memory hierarchy 2.3.2 Locality of cache memory 2.3.3 Cache memory operation 2.3.4 Cache memory system 2.3.5 Snoop
		2.4 Virtual memory technology	
			2.4.1 Characteristics of virtual memory 2.4.2 Mechanism of virtual memory 2.4.3 Virtual memory system
	3. I/O-related architectures		
		3.1 Outline of I/O interfaces	
			3.1.1 Interface driver (latch) performing signal level conversion 3.1.2 Interface LSI with LSI-based I/O and simple data conversion functions 3.1.3 Interface LSI with advanced data conversion functions controlling devices for specific uses
		3.2 I/O interface application technologies	
			3.2.1 Peripherals 3.2.2 Serial communication system 3.2.3 Parallel communication system 3.2.4 Analog interfaces 3.2.5 Communication interface devices 3.2.6 Communication standards 3.2.7 Transmission technology
		3.3 Peripheral device application technology	
			3.3.1 Parallel I/O LSI 3.3.2 Serial I/O LSI 3.3.3 Timer counter LSI 3.3.4 A/D and D/A conversion LSI 3.3.5 System LSI 3.3.6 Logic LSI with memory circuit mounted 3.3.7 DMD (Digital Micro Mirror Device) 3.3.8 Driver ICs (for liquid crystal panel control and other uses) 3.3.9 Cases of single-chip microcomputers

	4. Digital circuits and logic circuits		
		4.1 Electrical characteristics	
		4.2 Digital circuits	
			4.2.1 Gate circuits 4.2.2 Multiplexers 4.2.3 Three-state buffers 4.2.4 Flip-floppers 4.2.5 Encoders and decoders 4.2.6 Seven-segment decoders
	5. Peripheral device utilization technology		
		5.1 External storage devices	
			5.1.1 Memory Card, Memory Stick, SmartMedia, Multimedia Card, and SD Card 5.1.2 Micro Drive 5.1.3 DVD 5.1.4 CD-R and CD-RW 5.1.5 MO, HDD, FD, streamer, DAT, and others
		5.2 External I/O devices	
			5.2.1 Liquid crystal panels 5.2.2 LCD, LED, seven-segment LED, buzzer, vibrator, and others
		5.3 Power supplies	
			5.3.1 Characteristics of various power supplies and considerations 5.3.2 AC power supply 5.3.3 DC power supply 5.3.4 Batteries (shape, capacity, primary/secondary batteries, charge/discharge characteristics, and others)
	6. High-reliability design technology		
		6.1 Error detection and masking technology	
			6.1.1 Kinds and properties of errors 6.1.2 Error detection technology 6.1.3 Error correcting codes and error masking technology 6.1.4 Self-checking circuit
		6.2 Redundant configuration technology	
			6.2.1 Redundancy 6.2.2 Redundant configuration 6.2.3 Parallel redundant system 6.2.4 Stand-by redundant system 6.2.5 K-out-of-n redundant system 6.2.6 Considerations in redundant systems 6.2.7 RAID

		6.3 System recovery technology	
			6.3.1 Flow and level of recovery processing 6.3.2 Effects of maintenance in embedded systems 6.3.3 MTBF and MTTR
		6.4 Diagnosis technology	
			6.4.1 Diagnosis by software 6.4.2 Boundary scan
	7. Safety design technology		
		7.1 Effects from the surroundings	
			7.1.1 Noise control measures 7.1.2 Measures against EMI (electromagnetic interference) 7.1.3 Measures against EMC (electromagnetic compatibility) 7.1.4 Measures against short break and brownout in power supply (UPS) 7.1.5 Measures against electrostatic interference 7.1.6 Measures against surroundings such as temperature, humidity, and dust and against wrong operations by human beings
		7.2 Effects on the surroundings	
			7.2.1 Measures against EMI 7.2.2 Measures against electric leaks and electric shocks 7.2.3 Measures against heat generation and ignition 7.2.4 Measures against dangerous error output
		7.3 Safety standards and product liability	
			7.3.1 Safety standards 7.3.2 Product liability
		7.4 Failure analysis	
		7.5 Implementation technology	
		7.6 Encryption system	

Knowledge field	Major classification	Intermediate classification	Minor classification
D. Embedded-system development environment	1. System development environment	1.1 Development support tools	1.1.1 Document preparation tools
			1.1.2 Text editor
			1.1.3 Assembler, compiler, and linkage editor
			1.1.4 ICE and ICE monitor
			1.1.5 Configuration management tools
			1.1.6 Reverse engineering tools
		1.2 Platform	1.2.1 Platform
			1.2.2 Operating system
			1.2.3 Distributed development environment
		1.3 Development environment building method and its evaluation	1.3.1 Analysis of development work
			1.3.2 Development environment building method
			1.3.3 Methods of maintaining, managing, and utilizing the development environment
			1.3.4 Evaluation of the development environment
	2. Real-time system analysis and design techniques	2.1 Analysis techniques	2.1.1 Real-time structured analysis techniques
			2.1.2 Object-oriented analysis techniques
		2.2 Design techniques	2.2.1 Real-time structured design techniques
			2.2.2 Object-oriented design techniques
	3. Hardware design environment	3.1 Hardware description languages	3.1.1 Hardware description languages
			3.1.2 Hardware development design process
			3.1.3 Kinds and characteristics of hardware description languages
		3.2 Simulation technology	3.2.1 Trends in simulation technology
			3.2.2 Types of logic simulation methods
			3.2.3 Framework
		3.3 System LSI development techniques	3.3.1 Division of roles between ASIC users and semiconductor manufacturers
			3.3.2 ASIC development techniques
			3.3.3 FPGA design environment
			3.3.4 IP (intellectual property)

	4. Co-design		
		4.1 Outline of co-design	
			4.1.1 Objectives
			4.1.2 Effects
		4.2 Demarcation between the roles of hardware and those of software	
		4.3 Mutually grasping specifications and conditions	
		4.4 Review	
	5. Integrated development environment		
		5.1 Outline of the integrated development environment	
			5.1.1 Objectives
			5.1.2 Effects
		5.2 Types and characteristics of integrated development environments	
			5.2.1 Upstream development environment
			5.2.2 Downstream development environment
			5.2.3 Integrated development environment
		5.3 Configuration management	
			5.3.1 Significance of configuration management
			5.3.2 Documents subject to configuration management
	6. Standards related to embedded systems		
		6.1 Types of standards	
		6.2 Contents of standards	

Knowledge field	Major classification	Intermediate classification	Minor classification
E. Control theories and control technology in embedded systems			
	1. Control theories		
		1.1 Linear control system	
			1.1.1 Outline of the control system
			1.1.2 Configuration of the control system
			1.1.3 Transfer function
			1.1.4 Response characteristics
			1.1.5 Judging the stability of the control system
		1.2 Sample data control system	
			1.2.1 Outline of the sample data control system
			1.2.2 Pulse transfer function
		1.3 Non-linear control system	
			1.3.1 Non-linear control elements and analysis techniques
			1.3.2 Non-linear control system analysis techniques
	2. System control method		
		2.1 Sequence control	
			2.1.1 Basic configuration for sequence control
			2.1.2 Modeling techniques for sequence control
		2.2 Feedback control	
			2.2.1 Process control
			2.2.2 Servo control
		2.3 Various control methods	
			2.3.1 Advanced control (recent control method)
			2.3.2 Feedforward control
			2.3.3 Fuzzy control
	3. Distributed control		
		3.1 Control system sharing techniques	
			3.1.1 Distributed control method
			3.1.2 Configuration of the distributed control system
		3.2 Distributed control system	
			3.2.1 Function division type control system
			3.2.2 Resource sharing control system
			3.2.3 Micro-kernel and control software
		3.3 Self-distributing system	

	4. Mechatronics technology		
		4.1 System outline of mechatronics	
		4.2 Microprocessor control technology in mechatronics	
			4.2.1 Sequence control
			4.2.2 Servo control
		4.3 Control element technology in mechatronics	
			4.3.1 Sensor technology
			4.3.2 Actuator technology
		4.4 Software technology in mechatronics	
			4.4.1 Interrupt-based conventional method
			4.4.2 Method using a real-time operating system

Knowledge field	Major classification	Intermediate classification	Minor classification
F. Utilization of object-oriented technology			
	1. Characteristics of being object-oriented		
		1.1 Higher development efficiency	
		1.2 Development process	
			1.2.1 Spiral development
			1.2.2 Prototyping development
	2. Object-oriented analysis and design		
		2.1 UML	
		2.2 Use cases	
			2.2.1 Actors
			2.2.2 Events
			2.2.3 Use cases
		2.3 Scenario	
		2.4 Sequence diagram	
		2.5 Class diagram	
		2.6 Task division	
			2.6.1 Explicit task division
			2.6.2 Implicit task division
			2.6.3 Correspondence between task and class
		2.7 Task model	
		2.8 Messages	
		2.9 Phase diagram	
	3. Object-oriented programming		
		3.1 C++	
		3.2 Java	
	4. Embedded system development using Java		
		4.1 Characteristics of Java	
		4.2 Java2 platform	
		4.3 Embedded Java	
			4.3.1 Java2 MicroEdition (Java TV API, Java Auto API, and others)
			4.3.2 Jini
			4.3.3 Embedded Java
			4.3.4 HotSpot

Knowledge field	Major classification	Intermediate classification	Minor classification
G. New application fields of embedded systems			
	1. IT-related equipment		
		1.1 Digital home appliances	
		1.2 Personal data assistants	
			1.2.1 Portable telephone
			1.2.2 PDA
			1.2.3 PDA and access to the Internet
			1.2.4 PDA and electronic commerce
	2. Recognition technology		
		2.1 Image recognition	
			2.1.1 Pattern recognition
			2.1.2 Imaging technology
			2.1.3 Character recognition
			2.1.4 Handwritten character recognition
		2.2 Personal authentication	
			2.2.1 Fingerprint
			2.2.2 Retinal blood-vessel pattern
			2.2.3 Palm print
			2.2.4 Face
		2.3 Speech recognition and speech synthesis	
	3. Virtual reality technology		
		3.1 Human interface technology	
		3.2 Head mount display	
		3.3 Force feedback	
		3.4 Motion capture	

Information Technology Engineers Skill Standards
Technical Engineers (Embedded Systems) **Published on July 31, 2000**

Publisher Japan Information Processing Development Corporation
Central Academy of Information Technology
19th Floor, Time 24 Building, 2-45 Aomi, Koto-ku, Tokyo 135-8073
Tel +81 3 5531 0171 (key number)
Fax +81 3 5531 0170
URL http://www.cait.jipdec.or.jp