CHAPTER 14

From Components to Products: Gluing the Pieces Together

This chapter presents the CMMI® process areas of Product Integration, Verification, and Validation as a "triple." Here we show how their use guides projects from the building blocks developed during the Technical Solution phase to an integrated, verified, and validated set of product components that is ready for packaging and delivery. We also emphasize and reemphasize the importance of interfaces based on their own merit as well as from the CMMI® point of view.

The Product Integration process area addresses the integration of product components into either more complex components or subsystems, or into complete products. Although it is not wrong to think of product integration as a one-time assembly of the product components, it is normally an iterative process of assembling product components, evaluating them, and then assembling more product components until the complete product is assembled and the system is fully tested.

Product Integration can be a very deceptive process area to those who read its goals and practices as if all of the events were to be executed in sequential order. Its real value comes to the surface if one thinks about the Product Integration process area as a main software program that must be executed to assemble the product components, and Verification and Validation as its software subroutines that are called to perform peer reviews, testing, simulation, and other verification and validation activities.

The Integration Strategy

The performance of effective product integration involves the establishment and the maintenance of an integration sequence, the environment for performing the integration, and the development and use of integration procedures.

An integration and test strategy should be developed early in the project, concurrently with product development plans and specifications. Some life cycles require that the integration and test strategy be one of the very first documents developed for the project following the successful baselining of the allocated requirements.

During the establishment of the product integration strategy, the following 11 questions should be answered:

1. When will the product components be available?
2. Which ones are on the critical path?

3. What alternative integration sequences have been considered?
4. What work needs to be done to prepare and conduct the integration activities?
5. Who is responsible for each integration activity?
6. What resources will be required?
7. What schedule is to be met and what are the expectations?
8. Are the necessary procedures to be followed documented and in place?
9. Are any special tools required during the integration?
10. What must be included in the product integration environment?
11. What personnel skills are required for the individuals conducting and supporting the integration?

Other considerations include:

- What modules should be integrated first?
- How many modules should be integrated before integration testing starts?
- What order should be used to integrate the modules?
- Should there be more than one skeleton?
    - How is each skeleton defined?
    - Are there distinct build levels?
- How much testing should be done on each skeleton?

Alternatives for the order of product component integration include top-down product component integration, bottom-up product component integration, critical product components first, related functional product components first, as-available product components, and complete product component integration.

The Integration Environment

Of great importance is the establishment of the integration environment needed to support the integration of the product components. The requirements of an integration environment can vary widely depending on the product components that are integrated into the deliverable product. For contrast, imagine the integration environment needs for a software accounting system. Now compare these to the integration environment required for the development of a Boeing 747 aircraft. The space, safety concerns, assembly equipment, required personnel, tools, recording equipment, and so on, take a long time to plan and set up to support the integration for a Boeing 747. This integration environment is considerably different both internally to the systems themselves and externally to the users.

The environment may include test equipment, simulators, pieces of real equipment, and recording devices. It may be purchased or developed. The product integration environment may also include the reuse of existing organizational resources as long as reuse is planned for early enough in the project life cycle.

It is also quite possible for the integration, verification, and validation activities to share the same environment. One IT company in Asia and one in Central Europe set up "mirror" environments for their development, integration and systems test, and production functions. This company believed that it was cheaper in the long run to duplicate these environments with the same software and hardware rather than to worry about what effects partial or different environments might have on the development, implementation, and test results.

Product Integration Procedures

To carry out the integration sequence, product integration procedures must be developed or, if they already exist, reviewed for usefulness. These procedures may include guidance on the number of planned iterations and expected test results for each iteration. Criteria should also be established indicating the readiness of the product component for integration. Criteria may include the degree of simulation permitted for a product component to pass a test.

Readiness for Integration: Interface Requirements

One of the most important readiness criteria needed for product integration is the assurance that all of the product components are confirmed to be compliant with their interface requirements (Figure 14.1). Interface requirements drive the development of the interfaces necessary to integrate the product components.

The interface requirements may have been collected during the requirements elicitation phase and expanded during product architecture development, and they must be managed throughout the product life cycle. Interfaces are usually classified into three main classes: environmental, physical, and functional.

Typical categories for these classes include the following:

- Mechanical;
- Fluid;
- Sound;
- Electrical;

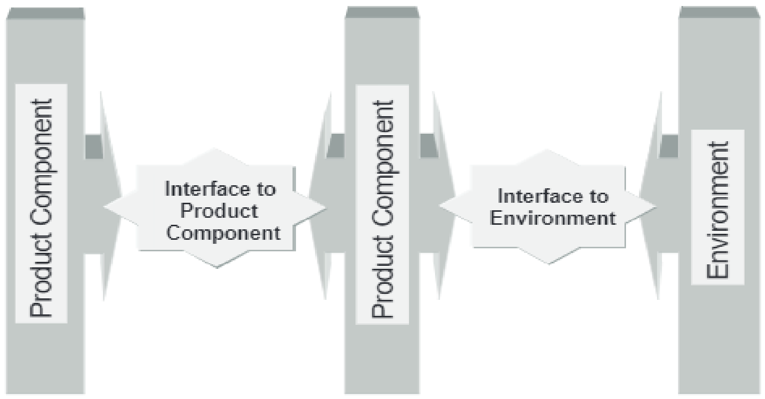

Figure 14.1 Ensuring interface compatibility.

- Climatic;
- Electromagnetic;
- Thermal;
- Message interfaces;
- Human–machine or human interface.

The interface descriptions defined during requirements evolution and the definition of alternative solutions must be placed under configuration management as one of the project's *most important* configuration items. Change requests to these interface descriptions must strictly follow the change control process. Many product integration problems arise from unknown or uncontrolled aspects of both internal and external interfaces. Effective management of product component interfaces helps ensure that implemented interfaces will be complete and compatible.

To ensure product component readiness for integration:

- Ensure that the product components are delivered to the product integration environment in accordance with the planned product integration strategy.
- Verify the receipt of each product component.
- Ensure that each product component meets its description.
- Check the configuration status of the product component against the expected configuration.
- Confirm that each product component is compliant with its interface requirements.

Ensuring interface compatibility and getting prepared for the assembly of the product components also includes *interface testing*. When organizations speak on the integration and systems testing phase, they rarely also speak of interface testing. The truth is that interfaces often do not get tested for one reason or the other and the cost and schedule pressure are typically listed as the culprits for this oversight. While teaching product integration during a CMMI® workshop, I often joke that systems testers are doing a bad job! This has the desired shock effect, and participants are curious as to how I am going to dig my way out of this one, especially those who have the job of systems testing. My response goes like this:

- Projects within organizations do not often carry out rigorous peer reviews on critical life-cycle components during the development stages.
- Unit testing is often left to the whims of the developers especially in software.
- Interfaces are not tested.
- The entire mess is passed along to systems test.
- The systems testers normally have half of the time requested for systems testing taken away from them.
- They try like crazy anyway and slog away at finding the defects that have been injected into the system.
- When they find defects and notify the developers, they are yelled at.

It therefore seems logical to come to the conclusion that systems testers do not do a good job. Of course this is a joke, but the point of making sure the interfaces are tested before systems testing of the product components is conducted is not a joke.

A reliability colleague of mine told me about a year ago that the systems testers for a major satellite project admitted they did not do a good job of testing the interfaces of components because it was too hard! The satellites only cost about $250 million.

Assembly of Product Components

Once the readiness criteria of the product components have been established, the actual process of integrating these product components can commence. This is normally expected to be done in an iterative fashion from the initial product components, through the interim assembly of product components, to the product as a whole. The readiness of the product integration environment should be ensured and the product components assembled according to the product integration sequence and integration procedures.

Evaluation of Assembled Product Components

Throughout the evolution of the product or product components, each interim state or final product state must be evaluated to show that they satisfy the functional, performance, and quality requirements. Actual product evaluation results should be compared against expected results and spot-checked by an independent party, such as quality assurance. The assembled product components must, therefore, be verified and validated according to the integration sequence and the verification and validation strategies.

Verification

Verification focuses not only on the final assembled product, but also includes verification of the intermediate work products against all selected requirements, including customer, product, and product component requirements. Verification methods address the technical approach to work product verification and the specific actions, resources, and environments that will be used to verify specific work products. Verification typically begins with involvement in the definition of product and product component requirements to ensure that these requirements are verifiable. The verification environment must be appropriate to support the verification method and may be shared with product integration and validation.

When verification activities are performed, the results must be documented. This includes the "as-run" verification method and the deviations from the strategies and procedures that were necessary during its execution that may be used later for process improvement purposes. During any verification activity, it is important to remember that the end goal is to ensure that each product component of a system satisfies its requirements.

System verification includes functional, physical, interface verification as well as word product verification. Functional verification includes *system performance testing,* which verifies performance with respect to the requirements, and *qualification testing,* which verifies system performance within its specified operational environment (e.g., temperature, vibration and shock, electromagnetic interference, electrical, flow system, and telecommunications).

Verification Techniques and Methods

Verification techniques ensure that the integrated product meets the specified requirements. In other words, it ensures that the project built the product right. Methods of verification include, but are not limited to:

- Inspections;
- Peer reviews;
- Audits;
- Walkthroughs;
- Analysis;
- Demonstrations;
- Comparisons;
- Path coverage testing;
- Simulations;
- Functional testing;
- Qualification testing;
- Factory testing;
- First article qualification;
- Bench testing;
- Load, stress, and performance testing;
- Operational scenario testing;
- Observations and demonstrations.

Verification methods commonly considered during system testing applied prior to packaging and delivery include:

- Load, stress, and performance testing;
- Functional decomposition-based testing;
- Operational scenario testing.

Systems Testing Versus Bench Testing

System testing can be defined as measuring a product component's performance while installed in a real or reasonable approximation of a functional system. Bench testing, in contrast, is defined as the insertion of a product component into a test loop where all of the variables can be independently controlled, measured, and recorded. Bench testing is also known as the process of characterizing the failure mode of a sample using the various bench equipment to stimulate the sample and measuring its response.

Validation

System validation is an end-to-end process that is needed to ensure that the completed and integrated system will operate as needed in the environment for which it was intended. It can be defined as a measure of customer satisfaction, given the customer's operational need and profile. System validation must always take into account the customer/end user during testing.

The development of validation procedures should include the test and evaluation of maintenance, training, and support services. As pointed out in Chapter 11, validation activities are performed as early in the product life cycle as possible, starting with the customer and product requirements. As the validation activities are performed, the "as-run" validation procedures should be documented and the deviations that occurred noted.

During validation it is important to remember to:

- Demonstrate that the maintenance tools are operating in the actual product,
- Verify in the field that support of the product is effective as specified by the customer (e.g., mean time to repair),
- Demonstrate adequate training of the products and services.

The final task during validation is to answer the question "Did the product perform as expected in its intended operational environment when used by the end users required to work in that environment?"

Validation procedures and criteria are established to ensure that the product or product component will indeed fulfill its intended use when placed in its intended environment. To validate a product means to demonstrate that you have built the right product. Validation methods should be selected based on their ability to demonstrate that user needs are indeed satisfied.

Validation Techniques and Methods

The CMMI® and other systems engineering sources offer examples of validation methods that include:

- Discussions with the users during the early phases of the life cycle;
- Prototype demonstrations;
- Simulation demonstrations especially with the end users present;
- Structured scenario testing;
- "Break it" testing to uncover unintended behaviors;
- Emulation of hardware components;

- Modeling (of hardware to validate form, fit, and function of mechanical designs);
- Reliability analysis;
- Functional demonstrations such as for the system, hardware, software, service documentation, and user interfaces;
- Pilots of training materials.

Acceptance Testing

Acceptance testing may satisfy the final verification and validation criteria. The purpose of acceptance testing is to confirm that a product or product component is ready for operational use. The acceptance test is performed for, or in conjunction with, someone else to demonstrate that the confidence is justified. The primary quality factors addressed are usability and reliability in order to answer the question "Will the product or product component support operational use?" Acceptance testing should also include the review of user documentation to verify its technical correctness compared to the system functionality, its completeness, and its ease of use.

Packaging and Delivery

Prior to packaging and delivering the product, the requirements, design, product, test results, and documentation are reviewed again to ensure that all issues affecting the packaging and delivery of the product are identified and resolved. Configuration audits are conducted prior to packaging and delivery to ensure the following:

- The product or product component that is included satisfies the customer and product requirements and all approved change requests and nothing more.
- The documentation that is to be delivered to the customer/end user matches the delivered product or product component.

Verification and validation results that have been conducted throughout the development life cycle are used as input to this final configuration audit.

The packaging requirements for some products can be addressed in their specifications and verification criteria, which are especially important when items are stored and transported by the customer.

Other important factors include environmental and stress conditions:

- Economy and ease of transportation (e.g., containerization);
- Accountability (e.g., shrink wrapping);
- Ease and safety of unpacking (e.g., sharp edges, strength of binding methods, childproofing, environmental friendliness of packing material, and weight).

Final delivery includes satisfying the applicable packaging requirements, preparing the operational site for the installation of the product, delivering the product and related documentation, and ensuring that it can be installed at the operational site.

Summary

The steps leading up to product delivery include establishing the integration environment, developing necessary integration procedures and criteria, ensuring that all product components meet their interface descriptions, integrating the product components according to the integration sequence, assembling the product components, and evaluating the product components, subsystems, and eventually the full system utilizing verification and validation techniques throughout the project life cycle as appropriate. Sequences of triples may be required to be carried out before the final product is assembled, verified and validated:

- (PI_1, VER_1, VAL_1),
- (PI_2, VER_2, VAL_2),
- (PI_3, VER_3, VAL_3),
- …,
- (PI_N, VER_N, VAL_N).

A delivered product is not considered to possess quality unless it can be shown that it satisfies all of the specified requirements and performs as anticipated in the operational environment by the customers or end users who have to use it.