



**PRECISION STUDIO**

A LEADER IN EFFECTIVE COMMUNICATION

**Project Management Methodology**

# **Integrate and Test SubPhase**



# Course Purpose



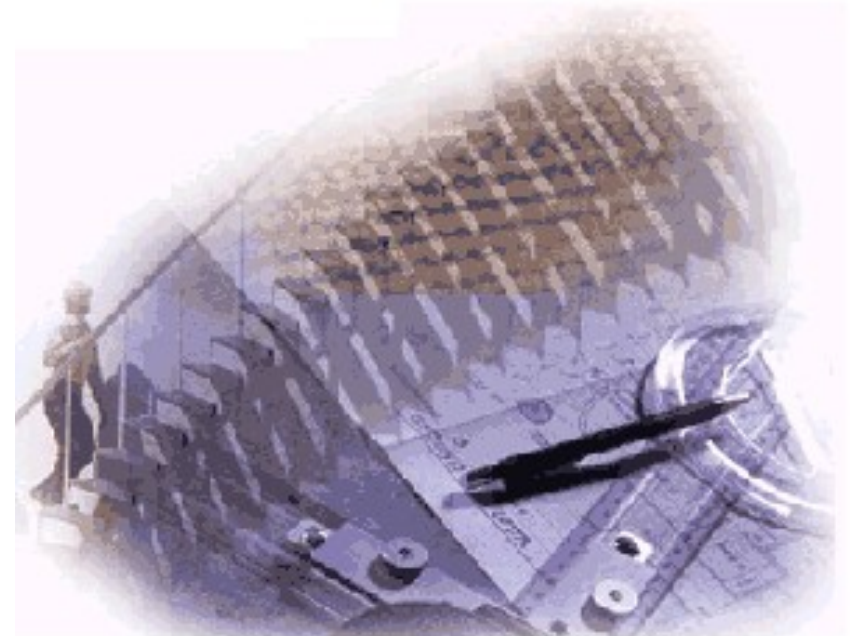
- Familiarize team members with the Integrate & Test Sub-Phase processes.
- Understand process flows, team member roles, and artifacts employed.
- Employ the course concepts to participate as a team member in the preparation of critical sub-phase deliverables.



# Integrate & Test Sub-Phase

## Agenda

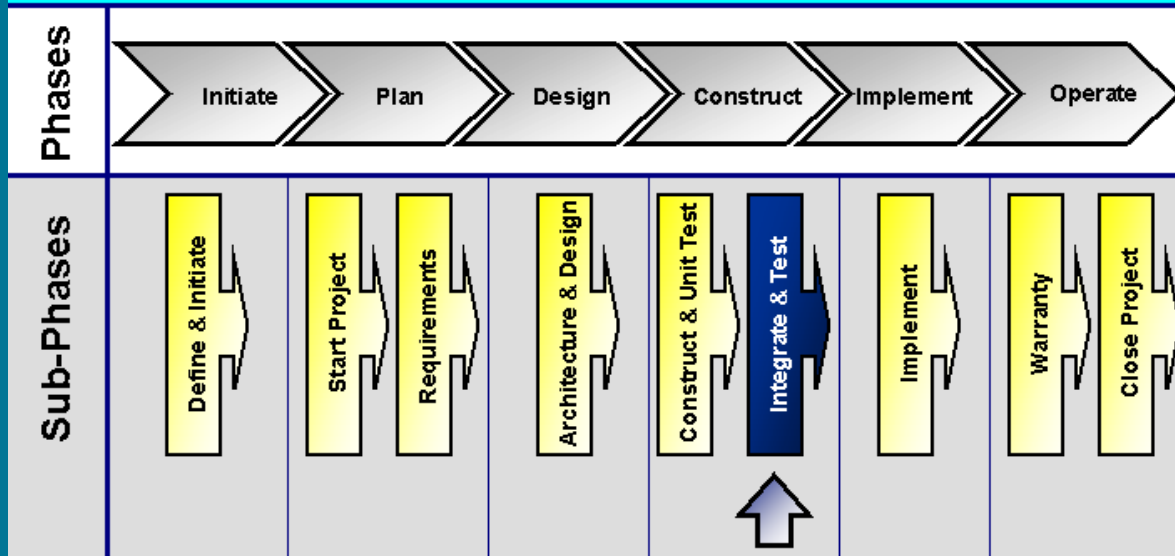
- Course Outline
- The Project Lifecycle
- Sub-Phase Purpose
- Team Members
- Inputs/Outputs
- Overview of Sub-Phase Processes





# The Project Life Cycle

## Phases and Sub-Phases of the Project Lifecycle



- A Project Solutions Methodology defines a standard project lifecycle.
- The Integrate & Test Sub-Phase is executed during the Construct Phase of the standard project lifecycle.



# What is Integration and Testing?

Integration is a process for combining pre-existing solution elements into an integrated environment. On a single platform, integrating software elements is a straightforward undertaking. However, with multiple platforms or environments, the challenge of integration becomes greater. Similarly, integration across multiple types of elements (software, data, organizational elements, and security) also can be challenging. The recommended approach to integration is, first, to iteratively assemble elements within one type (e.g., application elements), testing the integration of those solution "pieces" as each new element is added. Next, integration and testing occur iteratively across element types as each new element type is added to the solution.

Once the solution has been assembled from an IT perspective, successive types of *testing* (such as system, acceptance, and operational readiness testing) are performed. Each type of testing has different objectives and areas of emphasis. For example, system testing confirms that all IT system elements operate together as planned, perform as expected, and meet business functionality objectives. The emphasis in acceptance testing is to demonstrate that the solution meets business functionality, usability, and business process objectives.

The various testing phases are supported by a variety of testing *techniques*. An overview of these testing techniques is provided in the Techniques section of this Handbook. The type of technique used at any point in testing will be determined by the objective of the testing being done. In the Techniques section of this Handbook, testing techniques are cross-referenced to the phases of testing where they can be applied.



# What Makes for "Good" Integration & Testing?

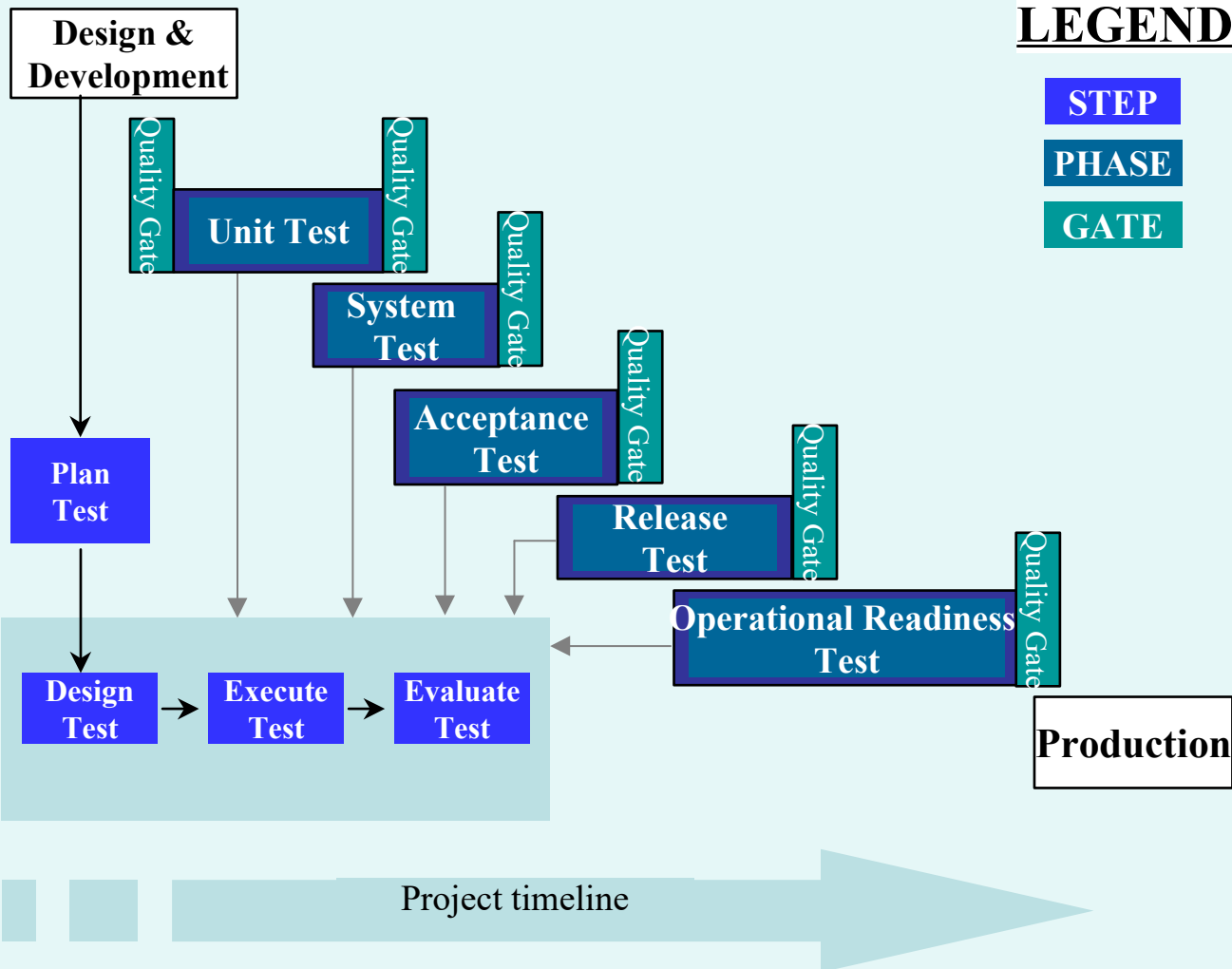
All of the effort expended during a skilled execution of the Initiate, Plan, and Design Phases of the lifecycle really begin to pay off during the Integrate & Test Sub-Phase. Ideally, this is a sub-phase in which there should be few surprises. After all, the deliverables developed in preceding phases and sub-phases are foundational elements for the execution of this sub-phase:

- Requirements (both functional and non-functional) are well documented and understood.
- Requirements (both functional and non-functional) have been mapped to design/development elements and to test cases.
- Test scenarios and test cases have driven design and the development of code.
- At each phase of the lifecycle, all aspects of the problem and the solution have been looked at from a variety of perspectives – business process, data, application, and technology, and organization.
- The development coordination process has ensured that the various teams involved in the project have consistently communicated and moved toward the common goal.
- Interface issues have been examined in detail.
- Solution elements have been thoroughly unit tested.

Thus, when integration is performed iteratively, any problems encountered should be related to interfaces between solution elements, and should be able to be readily identified, diagnosed, and cured. Once integration is complete, the execution of test cases (which are akin to "specifications" for the solution developed) should serve to demonstrate that the solution has been built as specified, rather than to highlight errors in the interpretation of the requirements.



# What Makes for "Good" Integration & Testing?



The diagram illustrates how the testing phases fits into the project life cycle phases.



# What Makes for "Good" Integration & Testing?

Testing occurs during a number of test phases – during unit test, which is performed as part of the Construct & Unit Test Sub-Phase; during integration, in which the successful assembly of solution elements is tested; during system test, in which the entire solution is tested as a cohesive whole; during acceptance test, in which the business community verifies that the solution meets their needs; and, during operational readiness testing, during which the complete solution is confirmed to be ready for implementation in an operational environment.

The input to each testing phase includes the most up-to-date solution elements and the results of previous testing, including identified and prioritized defects that have not been corrected. Note that not only are system elements of the solution tested, but the business process and organizational elements should be tested as well.

The decision regarding whether or not to correct a defect is based upon criteria such as time and resources available, prioritization of the defect (e.g. how important it is to key stakeholders to correct the defect), criticality of the defect (e.g. whether the defect causes the system to malfunction or fail), likelihood of occurrence of the defect (e.g. whether the defect will arise frequently or rarely in operational use), and downstream impact of the defect (e.g. whether the defect will cause an unacceptable downstream impact for the solution). These considerations are an important aspect of defect management.





# What Makes for "Good" Integration & Testing?



It would be impossible to over-emphasize the importance of well-managed migrations of system components from one environment/test phase to the next environment/test phase.

Whenever a migration occurs, all aspects of the target environment, including technical infrastructure (hardware and software), data structures, application elements, and tuning considerations should be re-confirmed as being as expected – including software / application versions, etc.

Otherwise, what might appear to be minor differences in the environment can potentially lead to spurious test results and excessive time can be lost in attempting to diagnose the reason for unexpected test results.

Alternatively, and perhaps more dangerously, tests might execute properly as a result of environmental variables, and system defects might not be caught.



# Who Uses Integrate & Test Deliverables?



Integrate & Test Sub-Phase deliverables are used by the Project Manager, the Project team, and the business representatives to confirm that the solution meets the objectives and requirements for the project.

When issues or errors are identified, this documentation serves as input to the error correction process.



# Integrate & Test – What Will You Accomplish?



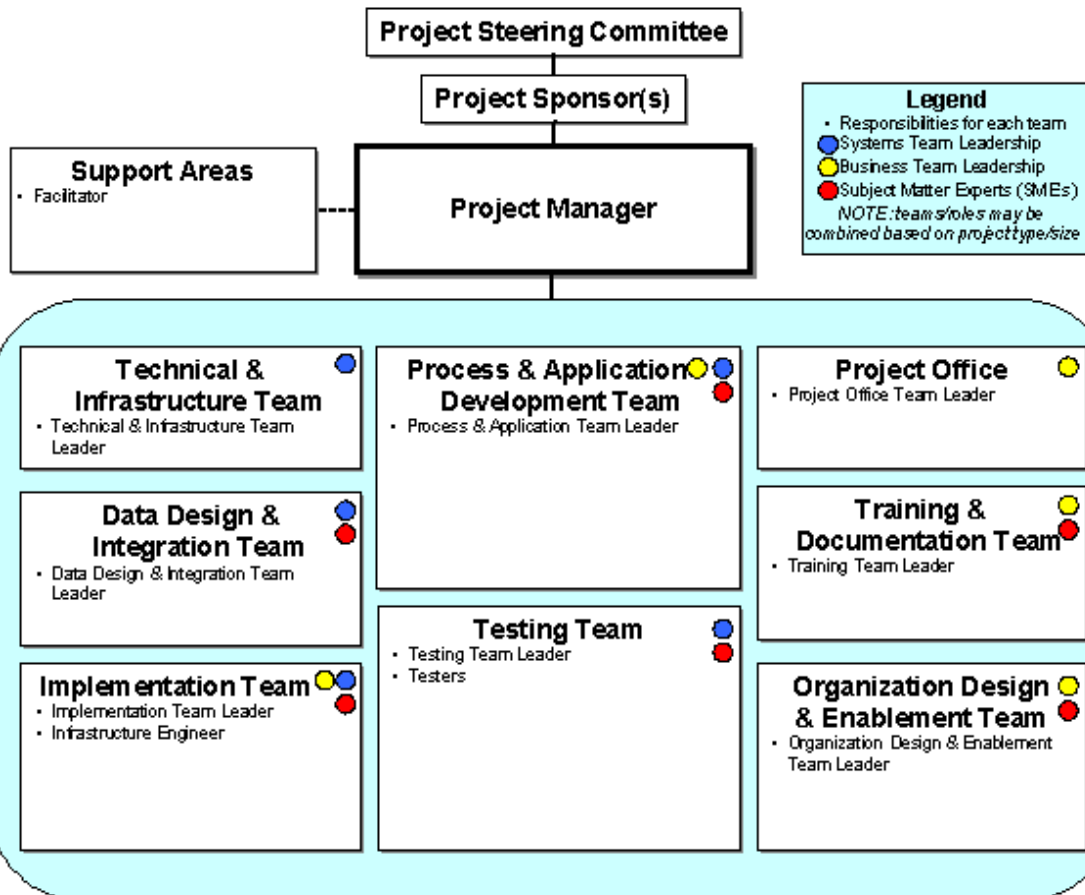
The purpose of the Integrate & Test Sub-Phase is to assemble all application functionality into a coherent whole and test the constructed functionality. It is critical to ensure that it is capable of achieving targeted business objectives. The elements that must be assembled include:

- Business process elements
- Software elements
- Data elements
- Organizational elements
- Security elements.

The key outputs from this sub-phase will combine with those from the Construct & Unit Test Sub-Phase completing the requirements for the Quality Gate Assessment. A successful review will move the project into the Implement Phase.



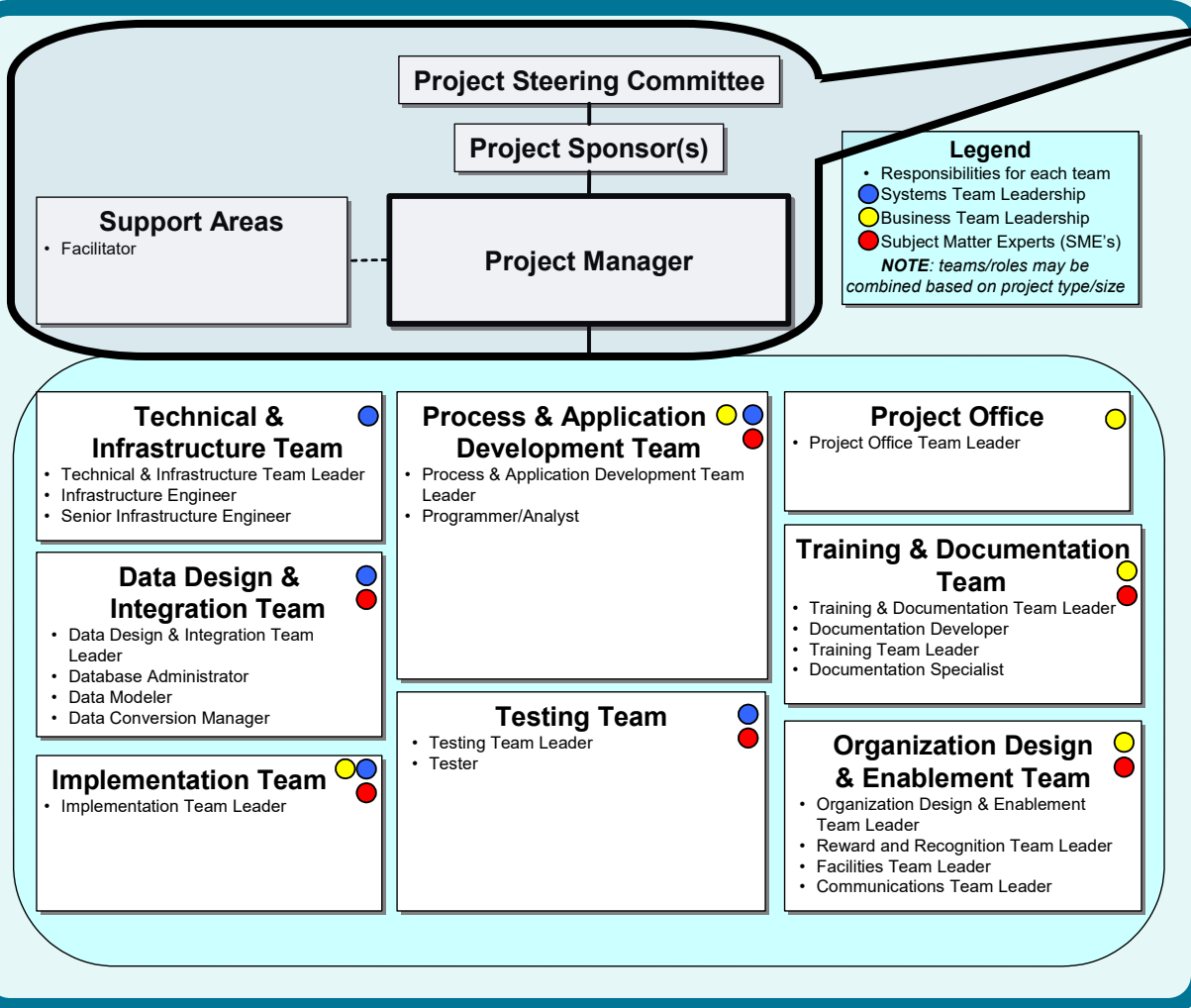
# Who Are Your Team Members?



- Individuals with the appropriate skills collaborating to drive a successful project.
- Team roles are subdivided according to functional requirements.



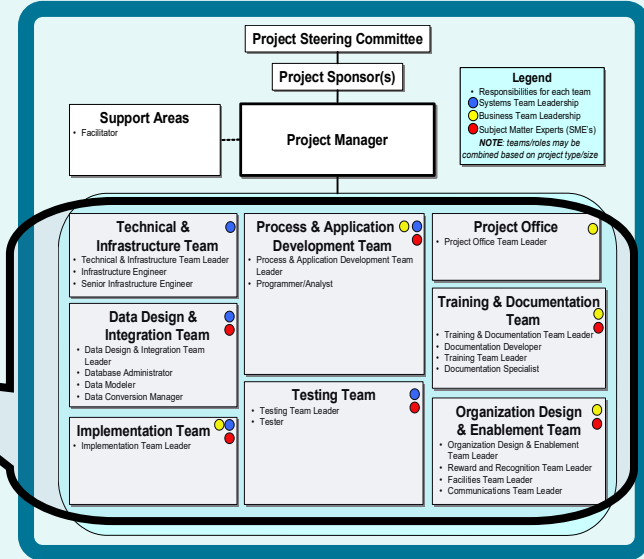
# The Management Team



- The Steering Committee provides senior management oversight and direction for a project.
- The Project Sponsor/Business Partner is the “owner” of a project.
- The Project Manager’s primary responsibility is to manage and coordinate day-to-day project activities.
- Support Areas provide Subject Matter Expertise



# The Support Teams



## The Project Office

Provides administrative support for Common Project Management Processes.

## The Implementation Team

The Implementation Team monitors the testing process in order to ensure that once implementation is to occur, all of the necessary elements and steps have been identified and documented.

## The Data Design & Integration Team

The Data Design & Integration Team is responsible for building and fine-tuning all data stores.

## The Technical & Infrastructure Team

The Technical & Infrastructure Team ensures that testing environments are prepared for each cycle of testing, and provides support for initializing environments as necessary to support iterations of the testing process.

## The Process & Application Team

Both system and business representatives are responsible for performing tests, documenting test results, and evaluating test results. Business representatives from this team are responsible for the business process and handoffs. They are also responsible for user acceptance testing, and will act as members of the testing team in that role.

## The Training and Documentation Team

The Training and Documentation Team is responsible for producing the educational materials, such as training and standard operating procedures, used by the Organization Design & Enablement Team to prepare testers.

## The Organization Design Team

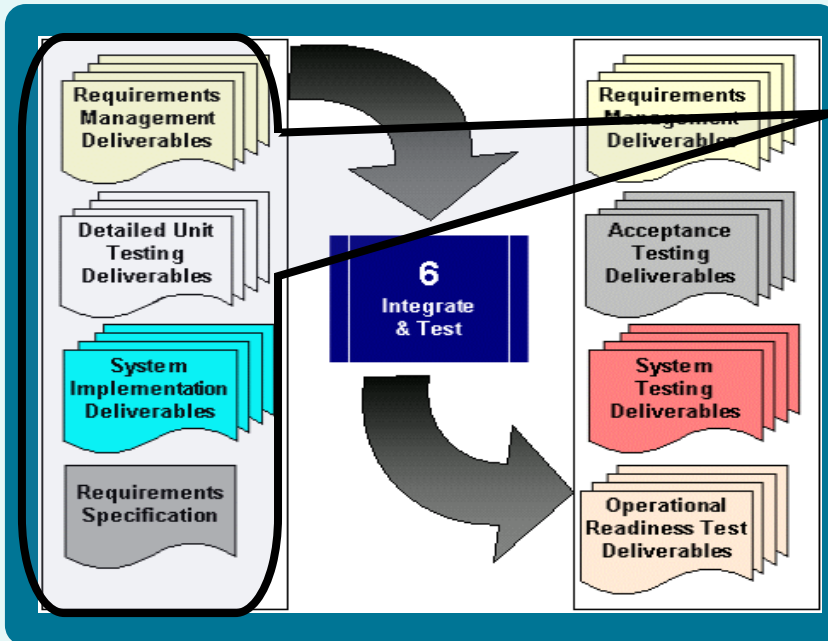
The Organization Design & Enablement Team is responsible for preparing testers to perform new or changed business processes using application testing techniques that interact with the process.

## The Testing Team

The Testing Team takes the lead, along with the Process & Application Team, in the execution of the test plan and the test cases during the Integrate & Test sub-phase. This team is responsible for identifying the testing strategy, the span of test scenarios, and testing flow. Also responsible for test tracking, documenting and evaluating test results, and reporting.



# What are the Critical Inputs?



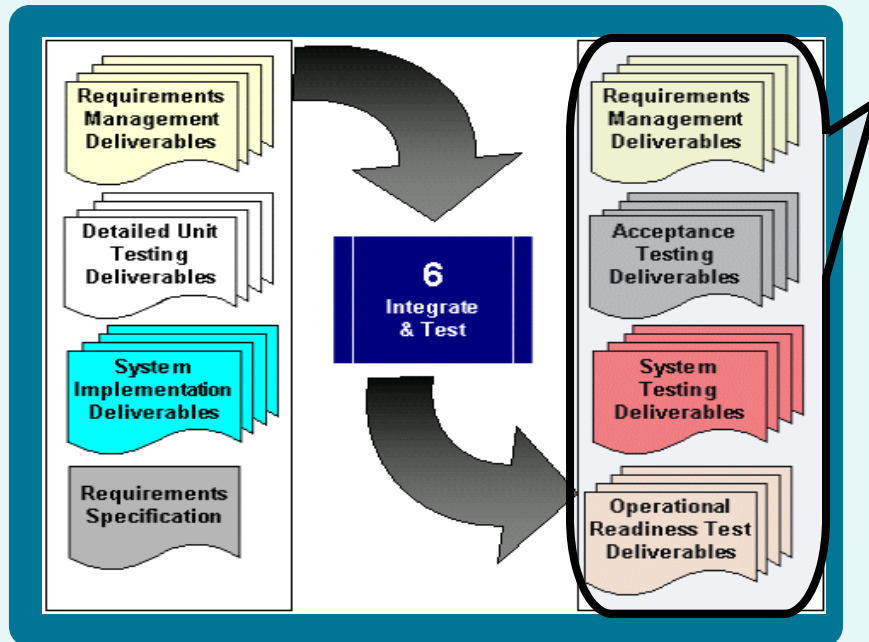
Key inputs to the Integrate & Test Sub-Phase consist of the following:

- **Requirements Specification:** A compilation of all business needs that must be met by the proposed solution. It includes, among other things, requirements for each project domain, i.e., process, application, data, technology, and organization.
- **Requirements Management Deliverables:** Created as part of the Requirements Sub-Phase, it includes an up-to-date Requirements Traceability Matrix, Requirements Management Strategy, Test Strategy, Work Force Enablement Strategy, System Implementation Strategy, and Post-implementation procedures for monitoring solution results during the Implement and Test Sub-Phase and after Implementation.
- **Detailed Unit Testing Deliverables:** This deliverable, created during the Construct & Unit Test Sub-Phase, includes unit test results and test scripts.
- **System Implementation Deliverables:** This deliverable, created during the Construct & Unit Test Sub-Phase, includes the required technology infrastructure, development database, database maintenance jobs/schedules, source code, standard operating procedures, initial program/module documentation and system run book for use during the Integrate and Test Sub-Phase.





# What Critical Outputs Will You Create?



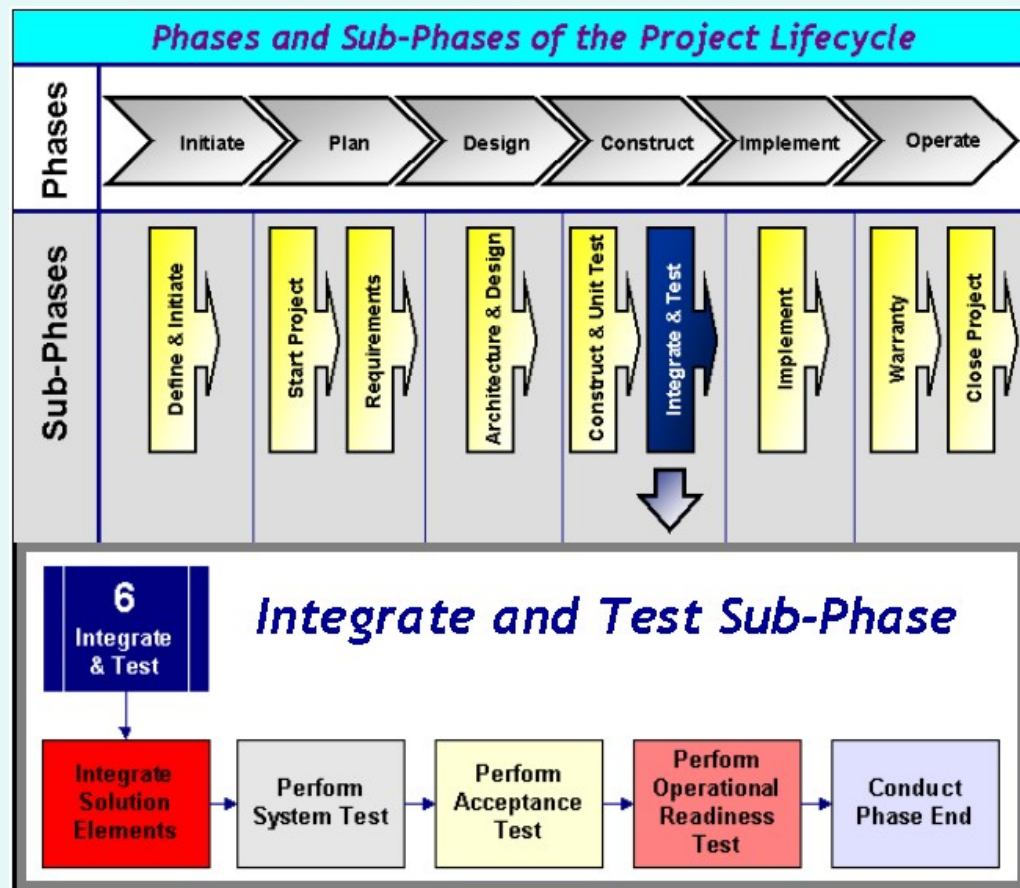
Key outputs of the Integrate & Test Sub-Phase include:

- **Fully Integrated Solution:** This deliverable is the fully tested, ready-to-implement solution that meets all of the requirements as specified – including requirements related to business process, technology, software applications, organizations, and work force enablement.
- **Detailed System Testing Deliverables:** This deliverable includes system test results, defect logs, and test scripts.
- **Detailed Acceptance Testing Deliverables:** This deliverable includes acceptance test results, defect logs, and test scripts.
- **Detailed Operational Readiness Test Deliverables:** This deliverable includes operational readiness test results, defect logs, and test scripts.
- **(Updated) Requirements Management Deliverables:** This deliverable, created as part of the Requirements Sub-Phase, includes an up-to-date Requirements Traceability Matrix, Requirements Management Strategy, Test Strategy, Work Force Enablement Strategy, System Implementation Strategy, and Post-implementation procedures for monitoring solution results continuing into Implementation.

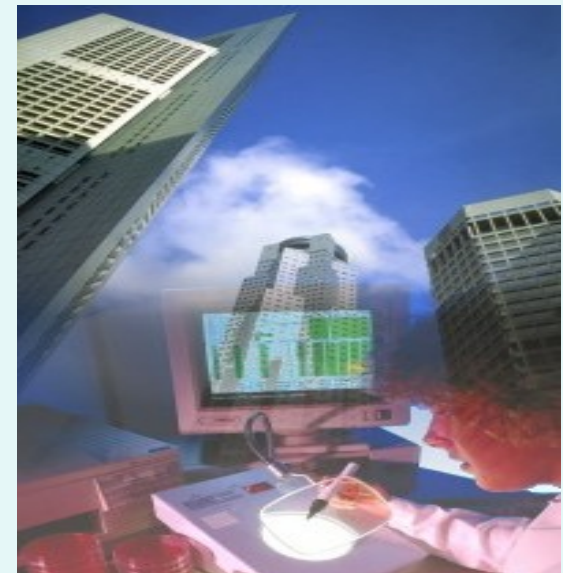




# How Does the Sub-Phase Breakout?

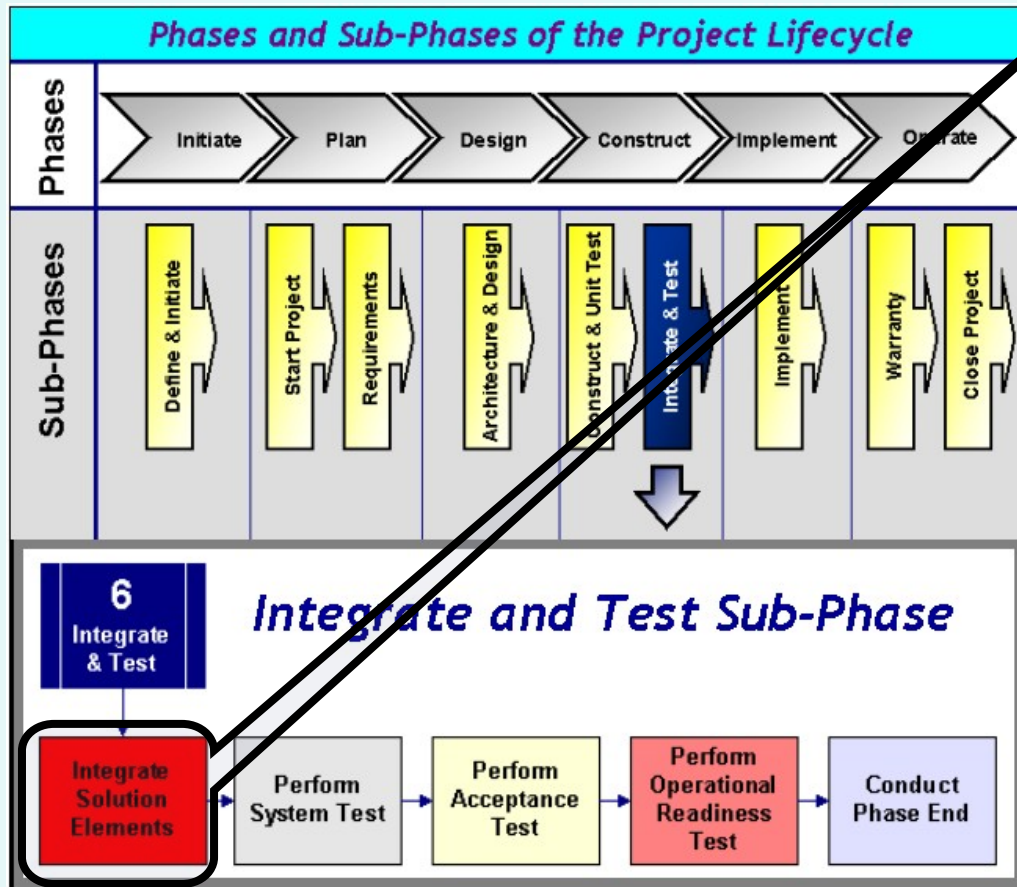


- The *Integrate & Test* sub-phase is broken into five processes.
- Depending on your role, you participate in one or more of the processes.





# Detail - Integrate Solution Elements



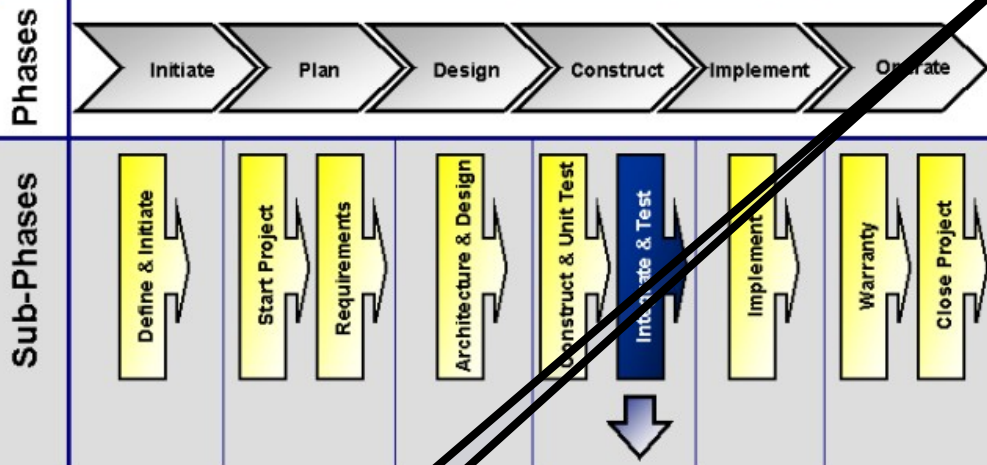
A variety of approaches may be taken for integration of the application elements for a solution iteration. On one extreme, all elements can be assembled at one time with testing occurring once all of the elements are assembled. Alternatively, elements can be assembled and tested iteratively. The advantage to the iterative approach is that the source of problems can be more easily identified.

The initial migration to the system test environment can occur all at once or may be done in phases. Should the application naturally parse into sub-applications, an initial phased migration from one test environment to the next may make good sense. Standards for versioning and version control (configuration management) are essential during the testing phases, since without such standards and control, system elements can very easily get "out of sync", and testing efforts may be wasted.



# Integrate Solution Elements

## Phases and Sub-Phases of the Project Lifecycle



6

Integrate & Test

## Integrate and Test Sub-Phase



## Example:

If two elements that have been thoroughly unit tested are assembled and a problem occurs, chances are very high that the problem has something to do with the interface between the elements – perhaps incorrect data is being fed from one element to another, data definitions are inconsistent, expected data is missing, etc.

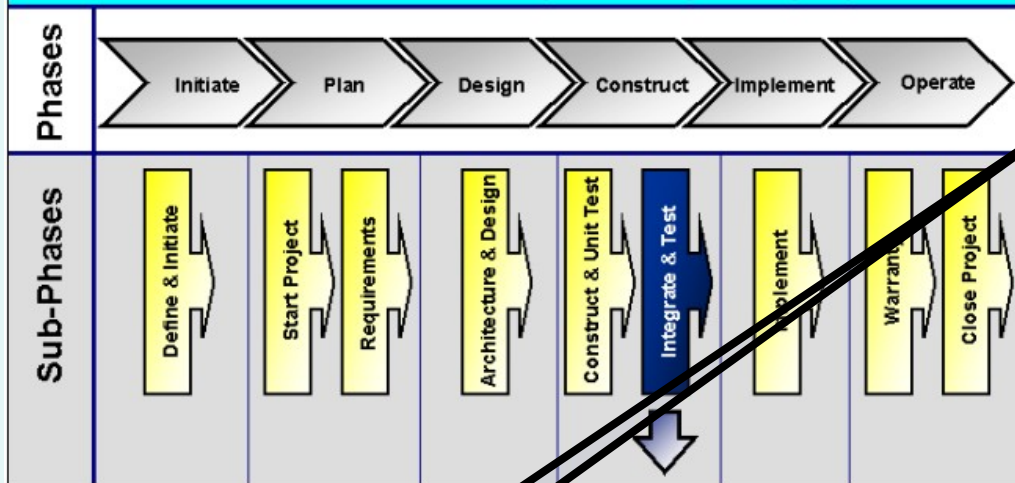
The perceived advantage of assembling all elements and then testing is that this approach is felt to be less time-consuming.

However, should difficult to diagnose problems arise, any savings gained in "all at once" integration of elements may be offset by time spent in tracking down the source of a difficult problem.



# Integrate Solution Elements

## Phases and Sub-Phases of the Project Lifecycle



This process is broken out into 4 sub-processes:

- Confirm Environment Readiness
- Integrate Elements
- Test Interfaces
- Migrate to Systems Test Environment/Phase

6

Integrate & Test

## Integrate and Test Sub-Phase

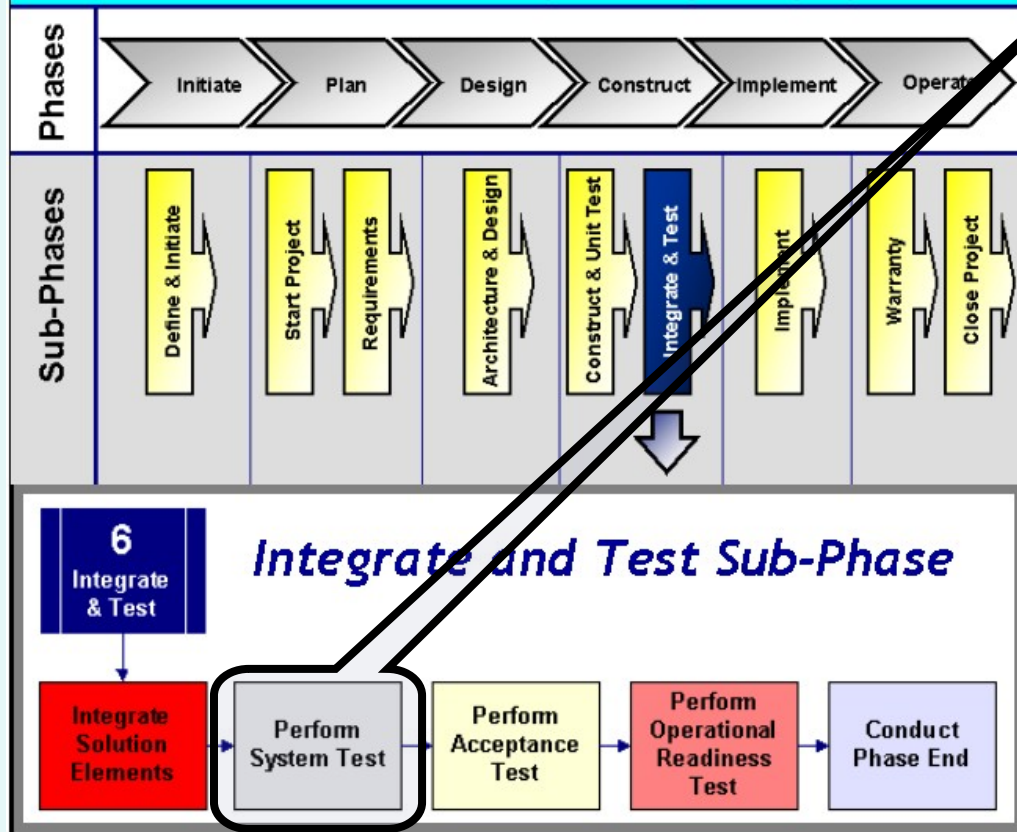






# Detail - Perform System Test

## Phases and Sub-Phases of the Project Lifecycle



System testing is the final opportunity for the technical community to ensure system quality before the business community begins the process of acceptance testing. Successful system testing demonstrates the following:

- Technical elements of the solution can operate together in an environment that (ideally) mirrors the production technical environment
- Business objectives have been met
- The solution is ready for acceptance testing by the business community.



# Perform System Test

## Inputs & Outputs

### Inputs

Requirements Specification  
Requirements Management Deliverables, including  
an up-to-date Requirements Traceability Matrix  
Detailed Unit Testing Deliverables  
System Implementation Deliverables

### Outputs

System Test Results  
System Test Evaluation  
Defect Log

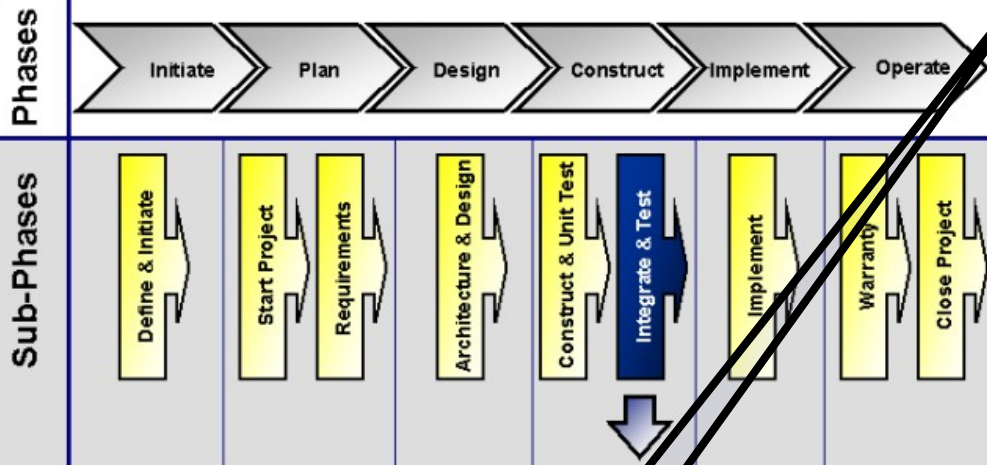
This process is broken out into 5 sub-processes:

- Confirm System Test Plan Environment Readiness
- Create/Confirm System Test Scripts
- Execute System Tests
- Evaluate System Test Results
- Migrate to Acceptance Test Environment/Phase

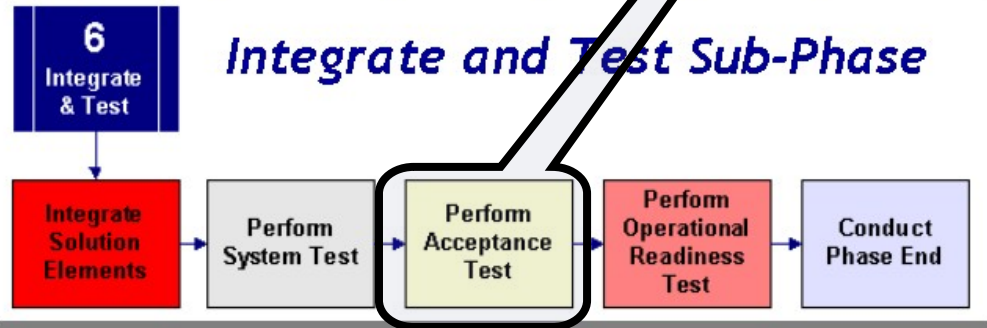


# Detail - Perform Acceptance Test

## Phases and Sub-Phases of the Project Lifecycle



## Integrate and Test Sub-Phase



Successful acceptance testing demonstrates that the solution meets the business community's requirements from the following perspectives:

- Functionality
- Regression
- Usability
- Business process
- Integration.

Standards for versioning and version control are essential during these testing phases. Without such standards and control, system elements can very easily get "out of sync", negating testing efforts.



# Perform Acceptance Test

This process is broken out into 5 sub-processes:

- Confirm Acceptance Test Plan
- Create/Confirm Acceptance Test Scripts
- Execute Acceptance Tests
- Evaluate Acceptance Test Results
- Migrate to Operational Readiness Test Environment/Phase

## Inputs & Outputs

### Inputs

Requirements Specification  
Requirements Management Deliverables, including  
an up-to-date Requirements Traceability Matrix  
Detailed Unit Testing Deliverables  
System Implementation Deliverables

### Outputs

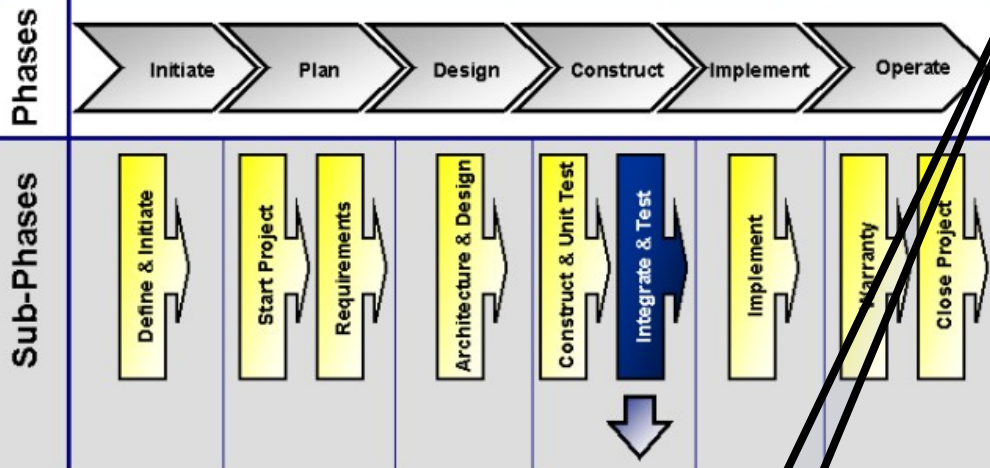
Acceptance Test Results  
Acceptance Test Evaluation  
Defect Log





# Detail - Perform Operational Readiness Test

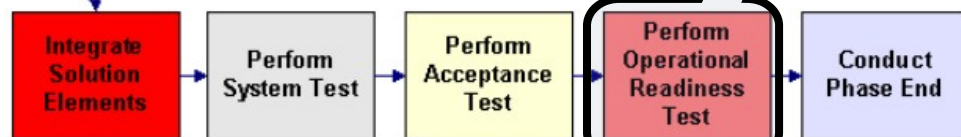
## Phases and Sub-Phases of the Project Lifecycle



6

Integrate & Test

## Integrate and Test Sub-Phase



Successful operational readiness testing demonstrates that the application meets operational and configuration requirements, including:

- Requirements for recovery
- Day-to-day operations
- Resource usage
- Configuration
- Interoperability
- Security
- Installation and backout.



# Perform Operational Readiness Test

## Inputs & Outputs

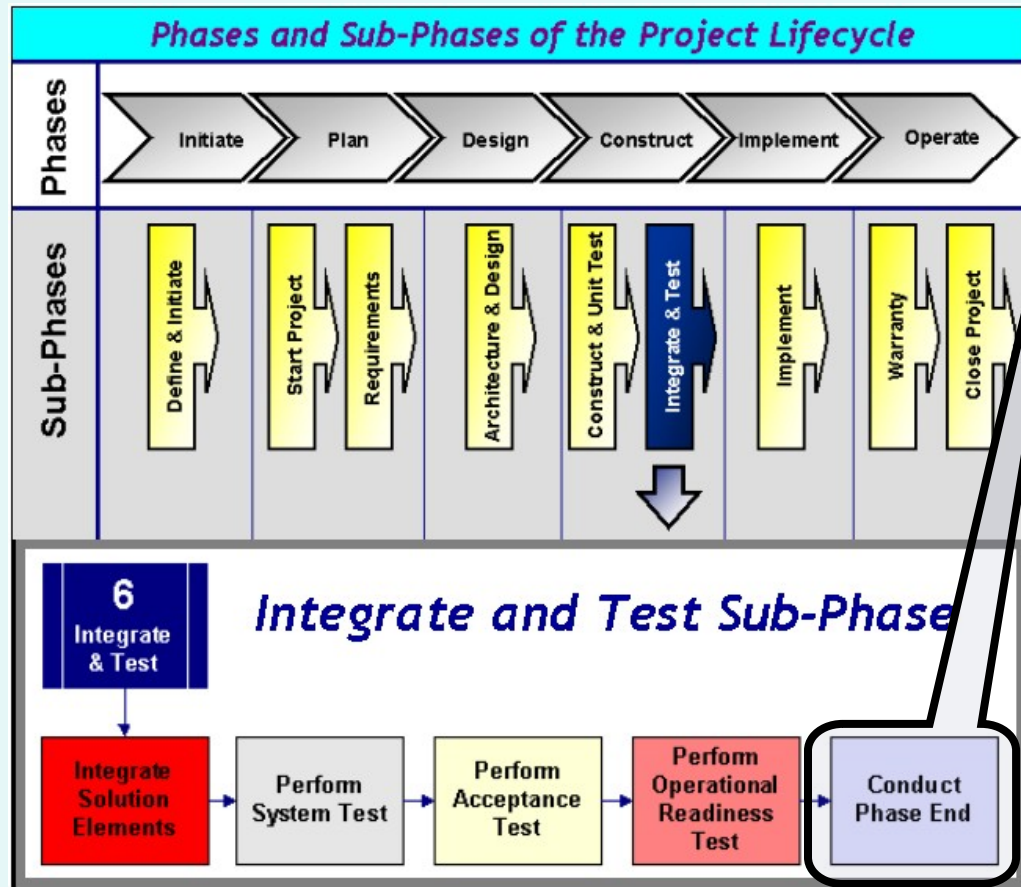
<b>Inputs</b>	Requirements Specification Requirements Management Deliverables, including an up-to-date Requirements Traceability Matrix Detailed Unit Testing Deliverables System Implementation Deliverables
<b>Outputs</b>	Operational Readiness Test Results Operational Readiness Test Evaluation Defect Log

This process is broken out into 4 sub-processes:

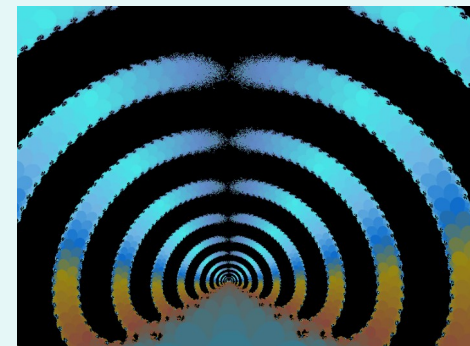
- Confirm Operational Readiness Test Plan
- Create/Confirm Operational Readiness Test Scripts
- Execute Operational Readiness Test
- Evaluate Operational Readiness Test Results



# Detail - Conduct Phase End



The standard project phases, as defined in the project lifecycle, incorporate phase end gates or "decision points" to allow for a senior management review. Just as a Business Case (with Cost Benefit Analysis) is presented to senior management for funding in the Define & Initiate sub-phase, additional validations of the project are performed at the subsequent phase end gates (e.g. Plan, Design, Construct).





# Conduct Phase End

To prepare for this review, a Project Manager would ensure the quality of their key deliverables produced during the sub-phase (see Quality Management), confirm the required Information Protection deliverables, update their project workplan for the next phase (see Planning & Estimating) and re-calculate the project costs and benefits based on progress to date, comparing to those originally approved in the Define & Initiate Sub-Phase.

These phase end gates or "decision points" enable Project Managers and leaders to the review and evaluate the project to ensure continued alignment with business objectives; review project financial and deliverables progress, and provide guidance and funding for the remaining phases.

After reviewing the results of the testing effort, if it is determined by management that at this point the project is ready to proceed, the project will move to the Implementation Phase. If management deems further testing/fixing needs to be done, the project team will perform the requested additional testing/fixing. Once the requested additional testing effort is finished, the project team will inform management and proceed to the Implementation Phase.



# Integrate & Test Summary

## ***Purpose***

- The purpose of the Integrate & Test Sub-Phase is to assemble all application functionality into a coherent whole and test the constructed functionality.

## ***Team Roles***

- Steering Committee, Project Sponsor, Project Manager, Support Areas, Project Office, Process and Application Team, Testing Team, Technical and Infrastructure Team, Data Design and Integration Team, Organization Design and Enablement Team, Training and Documentation Team, Implementation Team.

## ***Major Inputs***

- Requirements Management Deliverables
- Detailed Unit Testing Deliverables
- System Implementation Deliverables
- Requirements Specification

## ***Major Outputs***

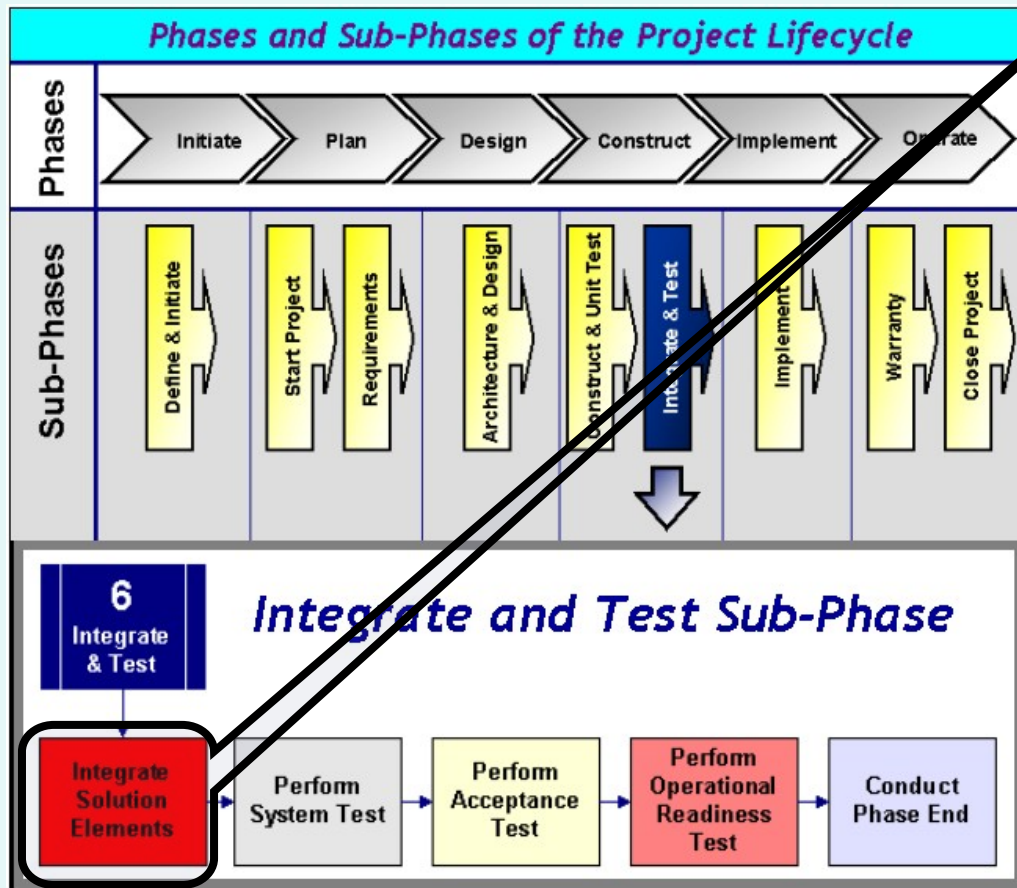
- (Updated) Requirements Management Deliverables
- Detailed Acceptance Testing Deliverables
- Detailed System Testing Deliverables
- Detailed Operational Readiness Test Deliverables
- Fully Integrated Solution

## ***Processes***

- Integrate Solution Elements
- Perform System Test
- Perform Acceptance Test
- Perform Operational Readiness Test
- Conduct Phase End



# Detail - Integrate Solution Elements



A variety of approaches may be taken for integration of the application elements for a solution iteration. On one extreme, all elements can be assembled at one time with testing occurring once all of the elements are assembled. Alternatively, elements can be assembled and tested iteratively. The advantage to the iterative approach is that the source of problems can be more easily identified.

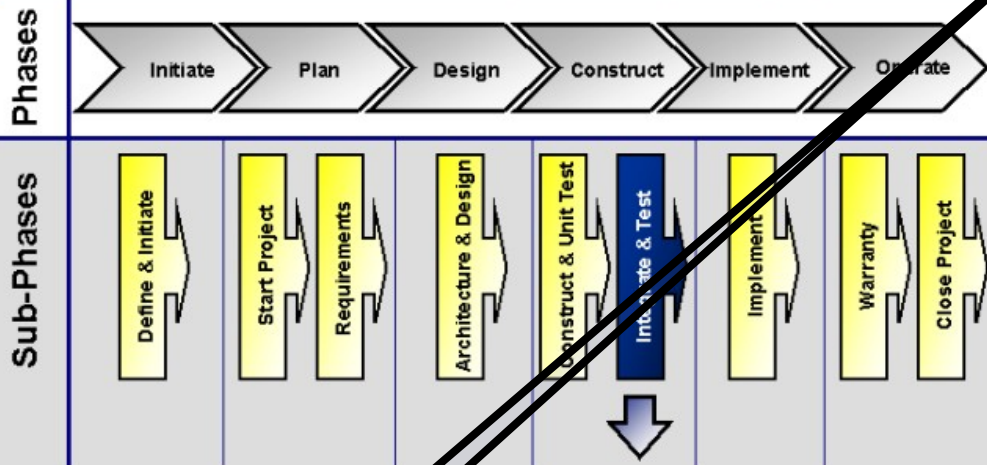
The initial migration to the system test environment can occur all at once or may be done in phases. Should the application naturally parse into sub-applications, an initial phased migration from one test environment to the next may make good sense. Standards for versioning and version control (configuration management) are essential during the testing phases, since without such standards and control, system elements can very easily get "out of sync", and testing efforts may be wasted.





# Integrate Solution Elements

## Phases and Sub-Phases of the Project Lifecycle



6

Integrate & Test

## Integrate and Test Sub-Phase



### Example:

If two elements that have been thoroughly unit tested are assembled and a problem occurs, chances are very high that the problem has something to do with the interface between the elements – perhaps incorrect data is being fed from one element to another, data definitions are inconsistent, expected data is missing, etc.

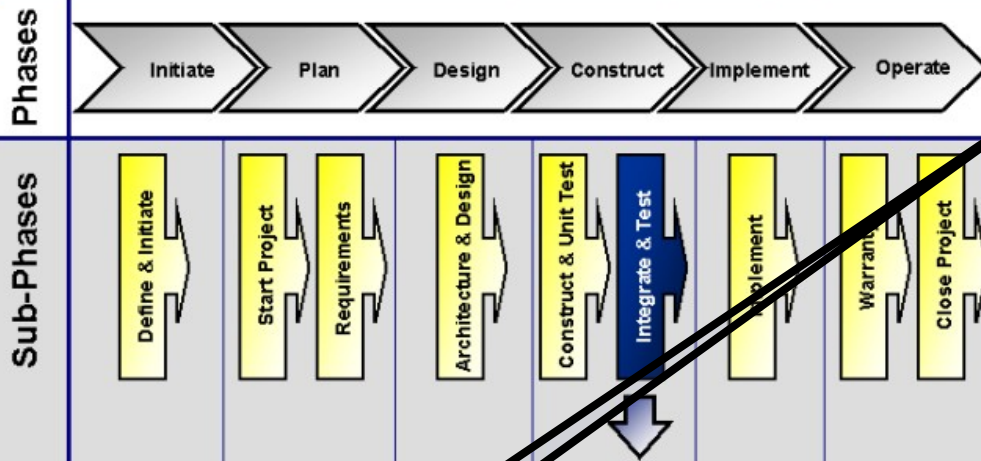
The perceived advantage of assembling all elements and then testing is that this approach is felt to be less time-consuming.

However, should difficult to diagnose problems arise, any savings gained in "all at once" integration of elements may be offset by time spent in tracking down the source of a difficult problem.



# Integrate Solution Elements

## Phases and Sub-Phases of the Project Lifecycle



This process is broken out into 4 sub-processes:

- Confirm Environment Readiness
- Integrate Elements
- Test Interfaces
- Migrate to Systems Test Environment/Phase

6

Integrate & Test

## Integrate and Test Sub-Phase





## Integrate Solution Elements

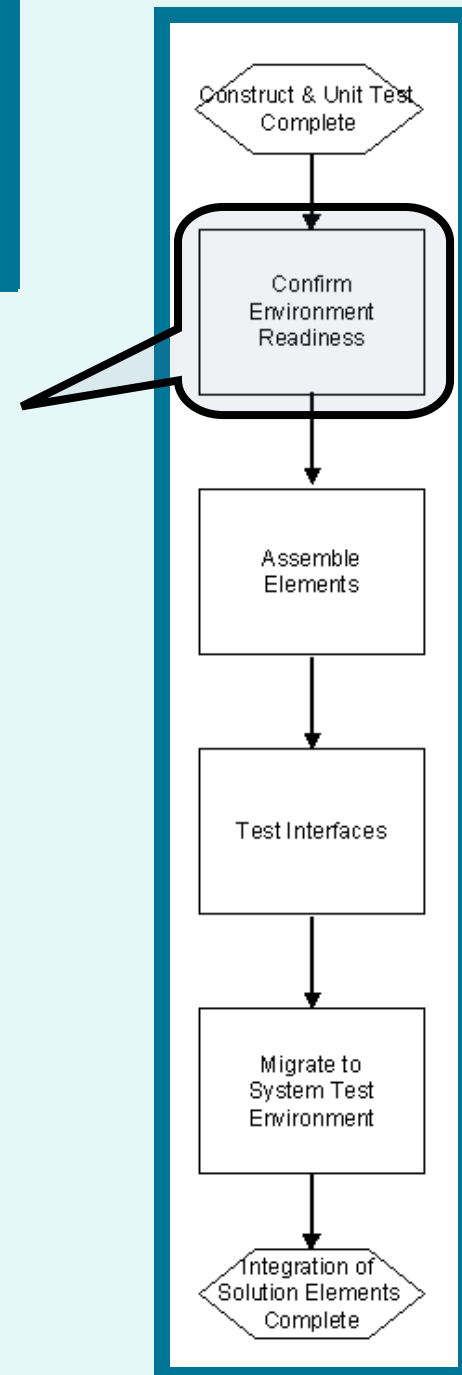
# Confirm Environment Readiness

Integration is best performed in a stable, controlled environment. In general, all required business process and infrastructure environments are built during the Construct & Unit Test Sub-Phase.

Nevertheless, the first activity performed during Integration is confirming the readiness of the environment in which integration activities will occur. This activity is especially important if integration activities will occur in the same technical environment as development activities.

The environment must be checked to ensure that it includes necessary utilities, data structures, converted data (if appropriate), and supporting applications. In addition, extraneous elements in the environment (which may have been used during development) must be removed from the environment in preparation for integration.

Once necessary environmental elements, *and only necessary environmental elements*, have been confirmed to be in the integration environment, application elements can be migrated and integration can begin.



# Confirm Environment Readiness

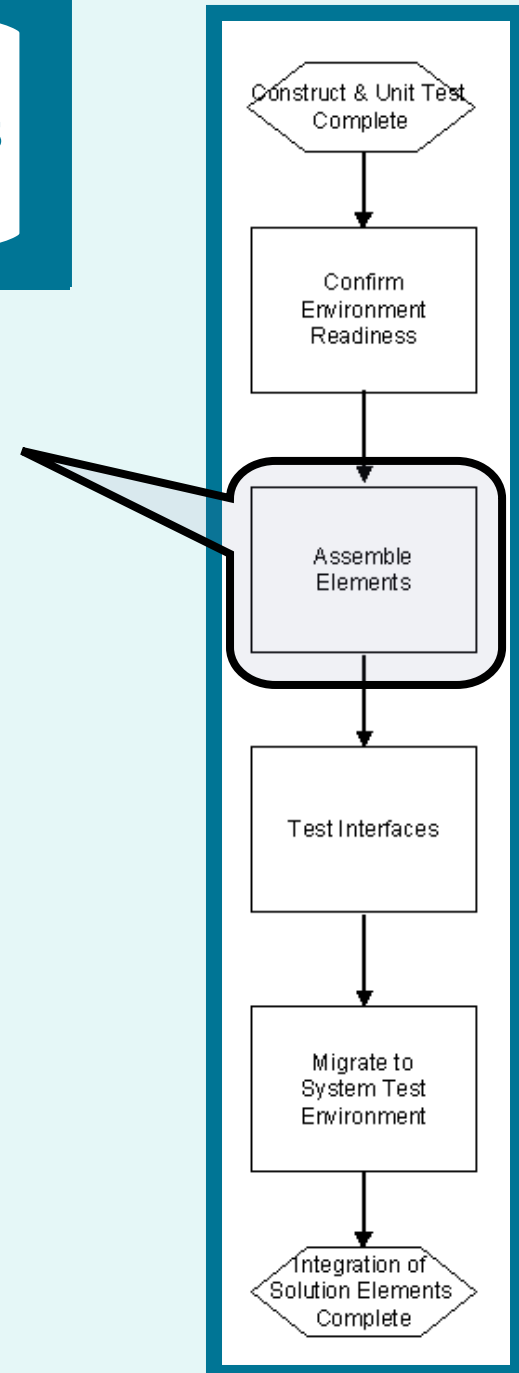
<b>Inputs &amp; Outputs</b>	
<b>Inputs</b>	Infrastructure Implementation Readiness Plan (IIRP) Database Technology Infrastructure
<b>Outputs</b>	(Confirmed) Technology Infrastructure (Confirmed) Database
<b>Roles &amp; Responsibilities</b>	
<b>Role</b>	<b>Responsibility</b>
Process and Application Team Leader	Executes this activity. Confirms that the integration infrastructure has been built and configured per requirements.
Process and Application Team	Supports this activity. Assists in confirming that all necessary application elements exist in the integration environment.
Data Design & Integration Team	Supports this activity. Assists in confirming that all necessary data elements exist in the application environment.
Infrastructure Engineer	Supports this activity. Assists in confirming that the environment is correctly configured.

## Integrate Solution Elements

# Assemble/Integrate Elements

During this activity, solution elements are integrated in order to assemble a fully working solution.

In most situations, the elements to be integrated are application elements. However, it is possible for any combination of solution elements to require integration and associated testing.



# Assemble/Integrate Elements

<b>Inputs &amp; Outputs</b>	
<b>Inputs</b>	Requirements Specification Requirements Management Deliverables, including an up-to-date Requirements Traceability Matrix Detailed Unit Testing Deliverables System Implementation Deliverables
<b>Outputs</b>	Integrated Solution
<b>Roles &amp; Responsibilities</b>	
<b>Role</b>	<b>Responsibility</b>
Process and Application Team	Executes this activity. Ensures that the application has been assembled completely.
Data & Integration Team	Supports this activity. Ensures that data elements support the integration of the application.

## Assemble/Integrate Elements

### **Example:**



A real-time business process flow might require both application and business process elements to be integrated. Imagine a customer service business process that involves several application elements to be integrated, allowing the customer service representative to access and enter data about a customer.

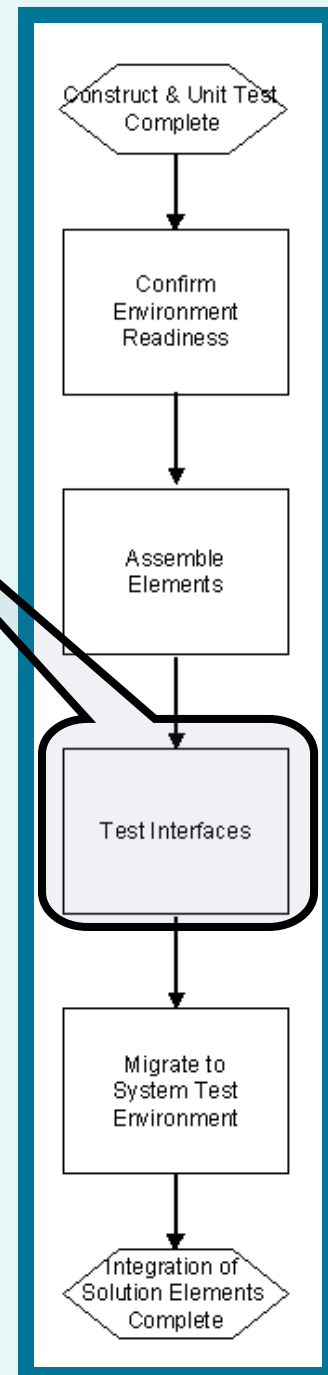
Then imagine that an intervening step requires the customer service representative to walk the customer through a process over the phone, but no application interaction is required. Subsequently, the customer service representative continues to access and update data about the customer.

This entire business flow, involving application and non-application components, must be integrated and tested.

## Test Interfaces

As an application is assembled, interfaces are tested to ensure that application elements will function as a cohesive unit.

This testing is not complete in terms of confirming functionality, but has the objective of ensuring that interfaces amongst application elements function as anticipated.



<b>Inputs &amp; Outputs</b>	
<b>Inputs</b>	Integrated Solution (under assembly)
<b>Outputs</b>	Integrated Solution
<b>Roles &amp; Responsibilities</b>	
<b>Role</b>	<b>Responsibility</b>
Process and Application Team	Executes this activity. Tests interfaces among application elements to demonstrate that elements interoperate as anticipated.
Data & Integration Team	Supports this activity. Ensures that data structures and elements support testing of application element interfaces.

## Test Interfaces



### **Example:**

If data were processed both on a mainframe and on a network, testing of the interfaces that pass data from one technical environment to another would occur during this process.



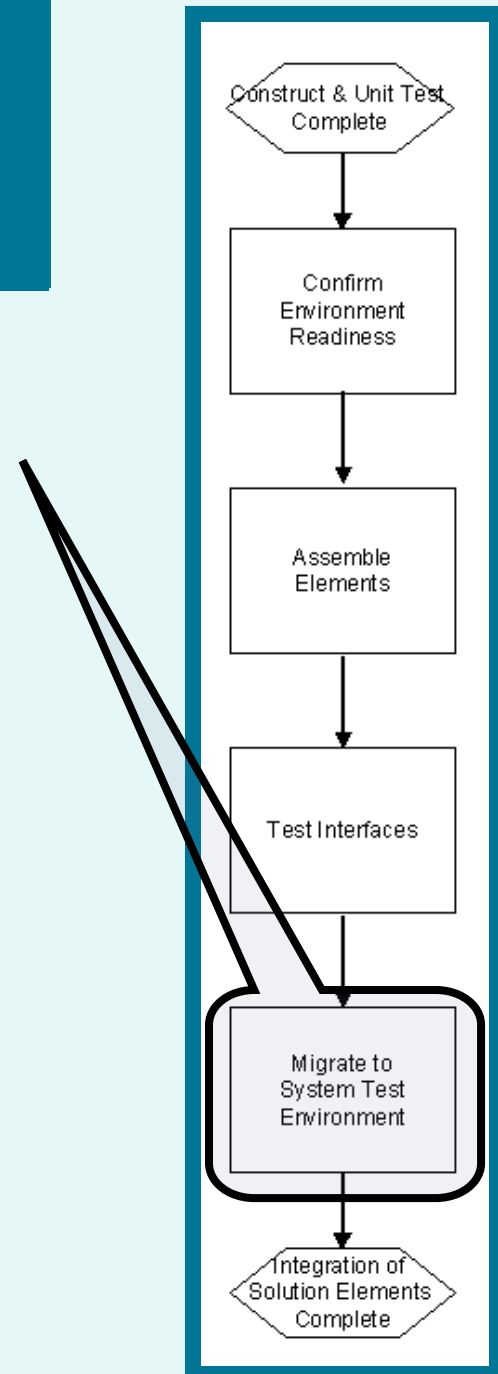
## Integrate Solution Elements

# Migrate to System Test Environment

Migration from the integration environment to the system test environment involves two major tasks – first, the environment is prepared for the migration, and, second, solution elements are actually migrated.

Since migrations between the integration and system test environments may occur iteratively, each iteration must start with preparation/confirmation that the system test environment contains the correct version of all necessary elements, no extraneous elements, the correct version of all necessary data and no extraneous data.

Only then should new or updated application elements be migrated into the system test environment.



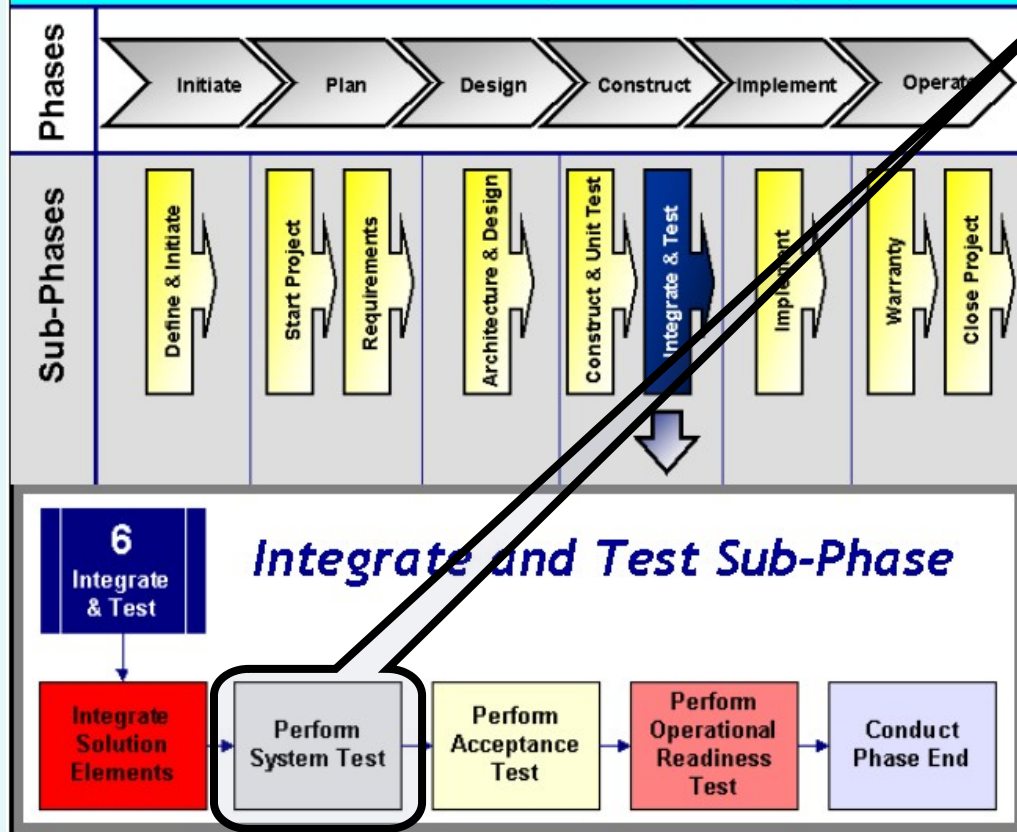
# Migrate to System Test Environment

<b>Inputs &amp; Outputs</b>	
<b>Inputs</b>	Integrated Solution
<b>Outputs</b>	Migrated Solution
<b>Roles &amp; Responsibilities</b>	
<b>Role</b>	<b>Responsibility</b>
Process and Application Team Leader	Executes this activity. Confirms that the state of the system test environment is appropriate for migration. Performs migration.
Process and Application Team	Supports this activity. Assists in confirming that all necessary application elements exist in the system test environment.
Data Design & Integration Team	Supports this activity. Assists in confirming that all necessary data elements exist in the system test environment.
Infrastructure Engineer	Supports this activity. Assists in confirming that the environment is correctly configured.



# Detail - Perform System Test

## Phases and Sub-Phases of the Project Lifecycle



System testing is the final opportunity for the technical community to ensure system quality before the business community begins the process of acceptance testing. Successful system testing demonstrates the following:

- Technical elements of the solution can operate together in an environment that (ideally) mirrors the production technical environment
- Business objectives have been met
- The solution is ready for acceptance testing by the business community.



# Perform System Test

## Inputs & Outputs

### Inputs

Requirements Specification  
Requirements Management Deliverables, including  
an up-to-date Requirements Traceability Matrix  
Detailed Unit Testing Deliverables  
System Implementation Deliverables

### Outputs

System Test Results  
System Test Evaluation  
Defect Log

This process is broken out into 5 sub-processes:

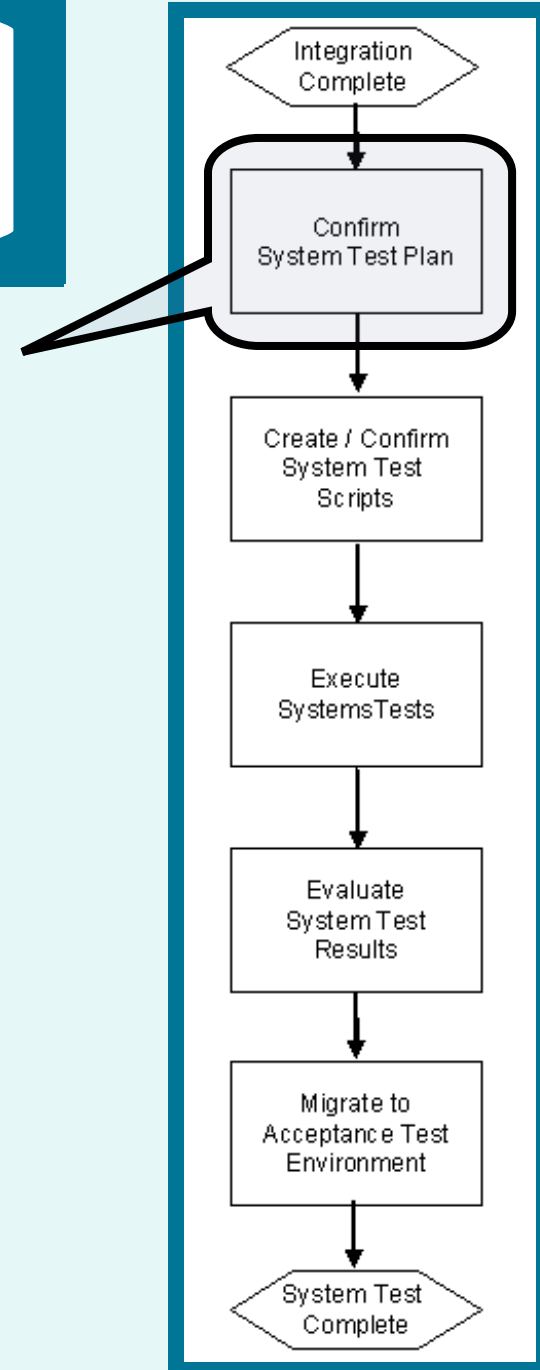
- Confirm System Test Plan Environment Readiness
- Create/Confirm System Test Scripts
- Execute System Tests
- Evaluate System Test Results
- Migrate to Acceptance Test Environment/Phase

## Perform System Test

# Confirm System Test Plan

The system test plan is created during the Architecture and Design Sub-Phase and is confirmed at the beginning of system test.

The completeness of the system test plan should be confirmed against the updated requirements traceability matrix.



## Perform System Test

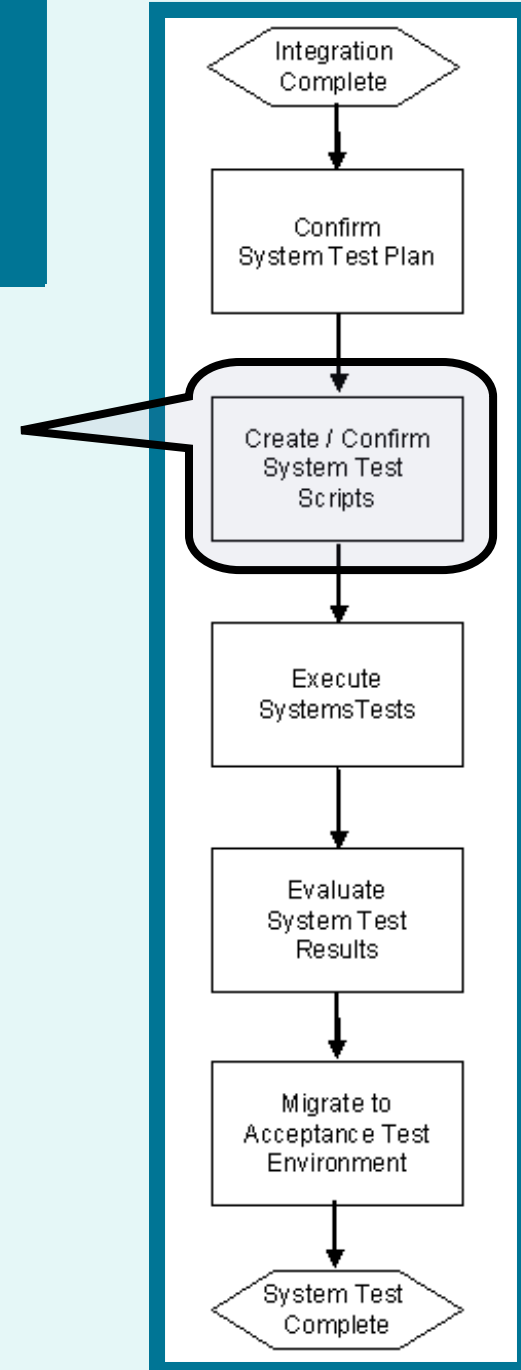
# Create/Confirm System Test Scripts

If system test scripts were created during an earlier sub-phase, the correctness of the scripts can simply be confirmed.

If system test scripts were not created earlier, they should be created from test cases and scenarios prior to the initiation of actual system testing. Test scripts should document all:

- Test conditions
- Data
- Expected results.

The completeness of the test scenarios and scripts should be validated against the updated requirements traceability matrix.



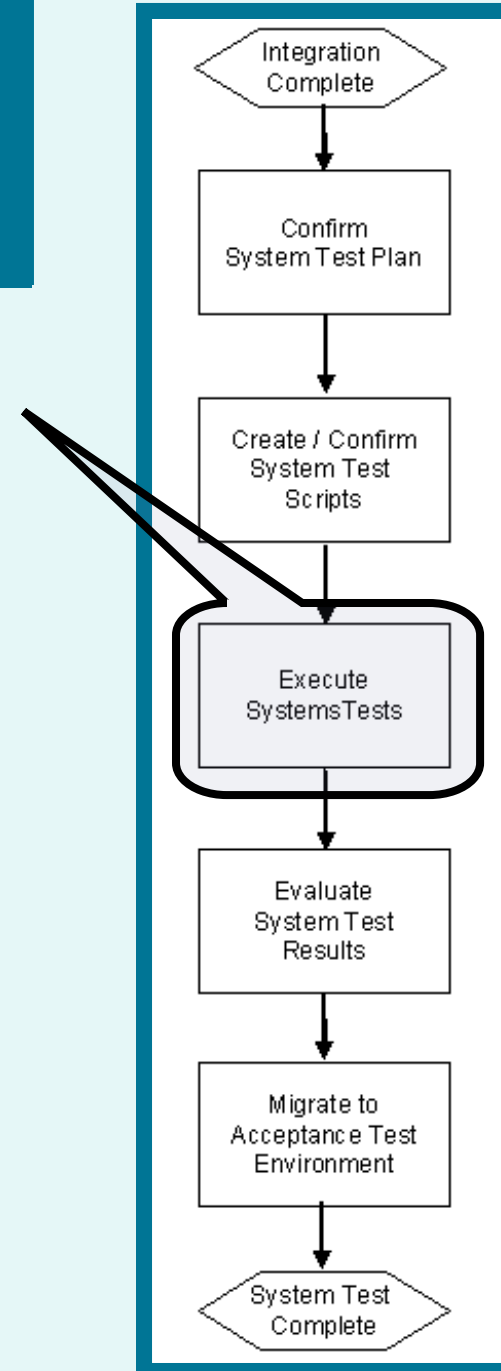
## Perform System Test

# Execute System Tests

Successful system testing demonstrates that:

- the technical elements of the solution can operate together in an environment that (ideally) mirrors the production technical environment,
- business objectives have been met, and
- the solution is ready for acceptance testing by the business community.

System testing is the final opportunity for the technical community to ensure quality in the system to be delivered before the business community begins the process of acceptance testing.



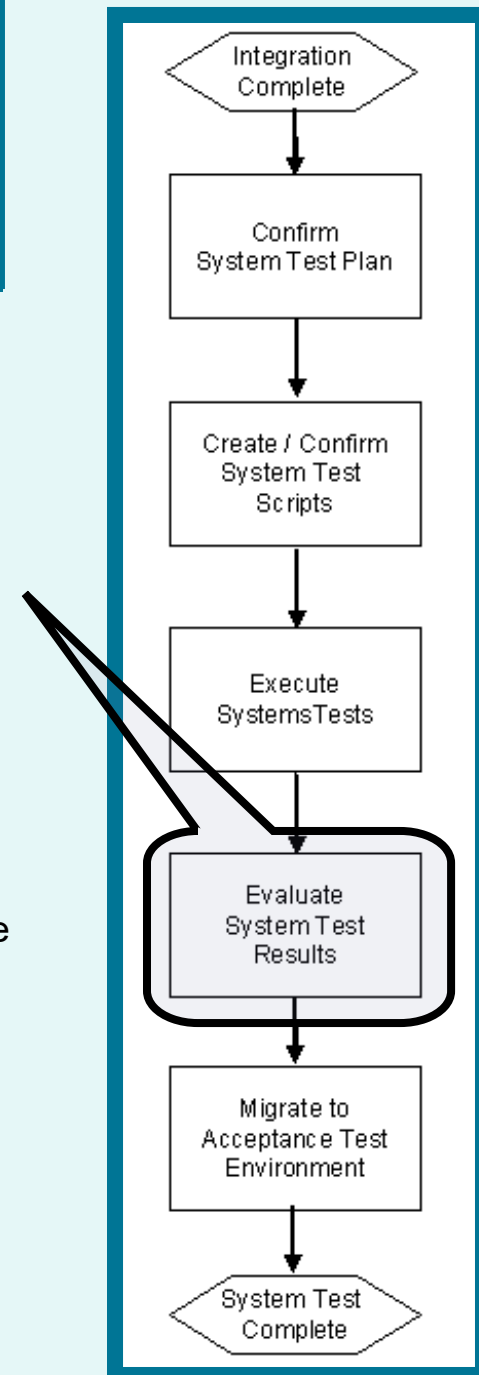
## Perform System Test

# Evaluate System Test Results

The Evaluate Test sub-process is initiated once the Execute Test sub-process is complete to determine if the test was successful. A successful test allows the application or solution iteration to move to the next test phase. An unsuccessful test requires errors to be fixed and the test re-executed. After the last phase of testing complete (i.e. Operational Readiness Testing), an Operations Certification is rendered to indicate whether the application or product is ready to move to production.

The Evaluate Test sub-process consists of the following steps:

- Compile and document test results.
- Evaluate the success of the test.
- Log defects for tracking purposes.
- Determine if the application/solution iteration passed or failed the test. Consider the acceptance criteria defined in the test plan for this test phase. For a failure, the reason for the failure must be determined (e.g. script error, software error).
- Identify opportunities to improve the testing process.
- Complete testing Quality Gates assessments.
- Initiate version control or change management, if application/solution iteration is ready for production.



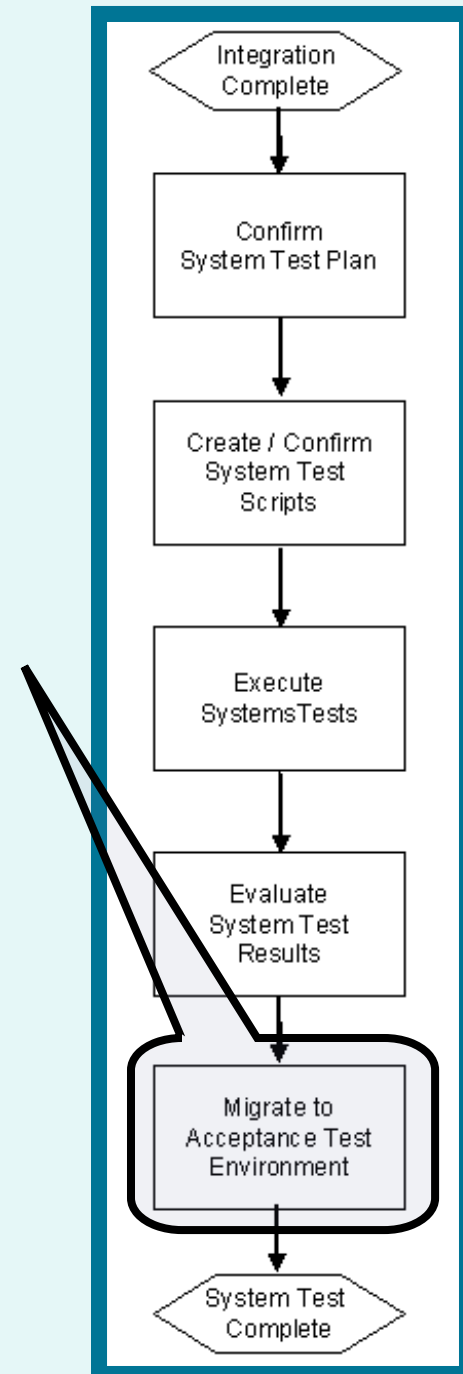


## Perform System Test

# Migrate to Acceptance Test Environment

Migration from the system test environment to the acceptance test environment involves three major tasks:

- First, the environment is prepared for the migration, and,
- Second, solution elements are actually migrated. Since migrations between the system test and acceptance test environments may occur iteratively, each cycle must start with preparation/confirmation that the acceptance test environment contains the correct version of all necessary elements, no extraneous elements, the correct version of all necessary data and no extraneous data. Only then should new or updated application elements be migrated into the acceptance test environment.
- Third, the user acceptance testers must be fully prepared to do user acceptance testing. Process / procedures aids must be prepared, and user acceptance testers must be fully trained in new and changed business processes so that they can test all elements of the solution adequately.



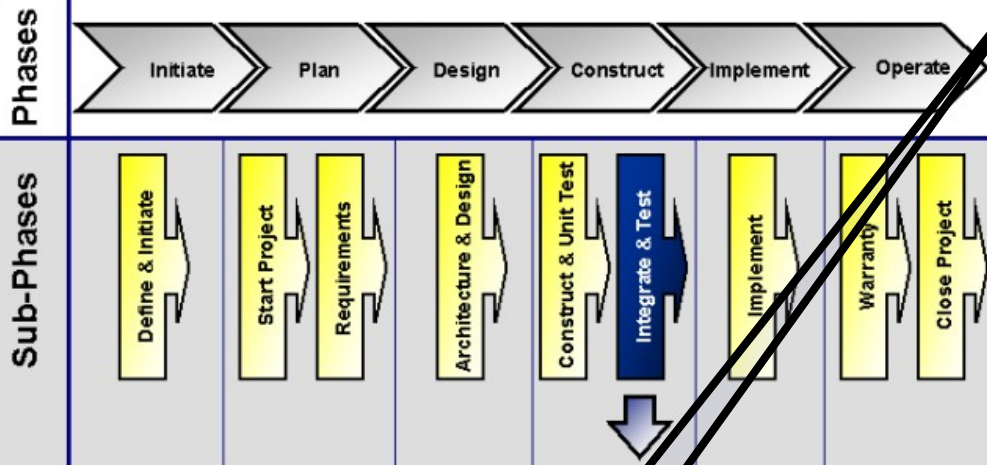
# Migrate to Acceptance Test Environment

<b>Inputs &amp; Outputs</b>	
<b>Inputs</b>	Integrated Solution
<b>Outputs</b>	Migrated Solution
<b>Roles &amp; Responsibilities</b>	
<b>Role</b>	<b>Responsibility</b>
Process and Application Team Leader	Executes this activity. Confirms that the state of the acceptance test environment is appropriate for migration. Performs migration.
Process and Application Team	Supports this activity. Assists in confirming that all necessary application elements exist in the acceptance test environment.
Data Design & Integration Team	Supports this activity. Assists in confirming that all necessary data elements exist in the acceptance test environment.
Infrastructure Engineer	Supports this activity. Assists in confirming that the environment is correctly configured.

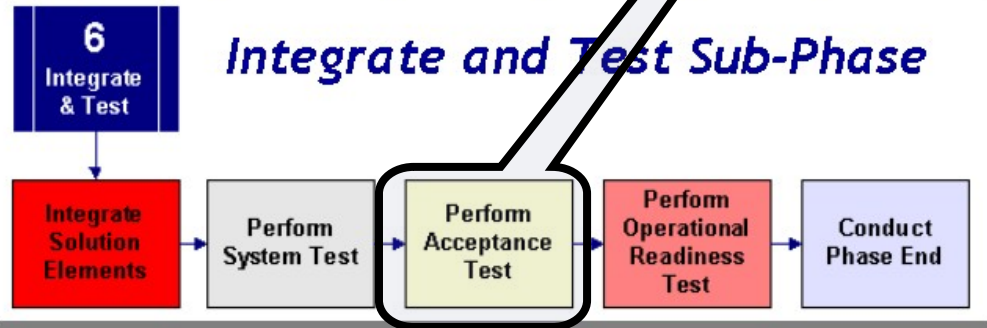


# Detail - Perform Acceptance Test

## Phases and Sub-Phases of the Project Lifecycle



## Integrate and Test Sub-Phase



Successful acceptance testing demonstrates that the solution meets the business community's requirements from the following perspectives:

- Functionality
- Regression
- Usability
- Business process
- Integration.

Standards for versioning and version control are essential during these testing phases. Without such standards and control, system elements can very easily get "out of sync", negating testing efforts.



# Perform Acceptance Test

## Inputs & Outputs

Inputs & Outputs	
<b>Inputs</b>	Requirements Specification Requirements Management Deliverables, including an up-to-date Requirements Traceability Matrix Detailed Unit Testing Deliverables System Implementation Deliverables
<b>Outputs</b>	Acceptance Test Results Acceptance Test Evaluation Defect Log

This process is broken out into 5 sub-processes:

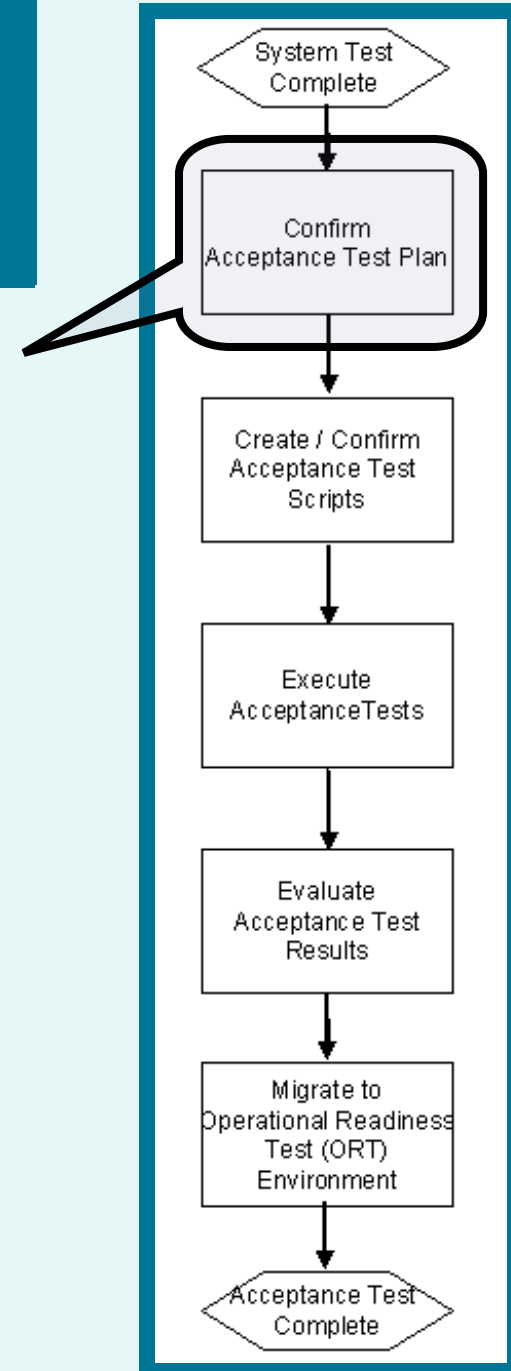
- Confirm Acceptance Test Plan
- Create/Confirm Acceptance Test Scripts
- Execute Acceptance Tests
- Evaluate Acceptance Test Results
- Migrate to Operational Readiness Test Environment/Phase

**Perform  
Acceptance  
Test**

## Confirm Assessment Test Plan

The acceptance test plan is created during the Architecture and Design Sub-Phase and is confirmed at the beginning of acceptance test.

The completeness of the acceptance test plan should be confirmed against the updated requirements traceability matrix.



## Perform Acceptance Test

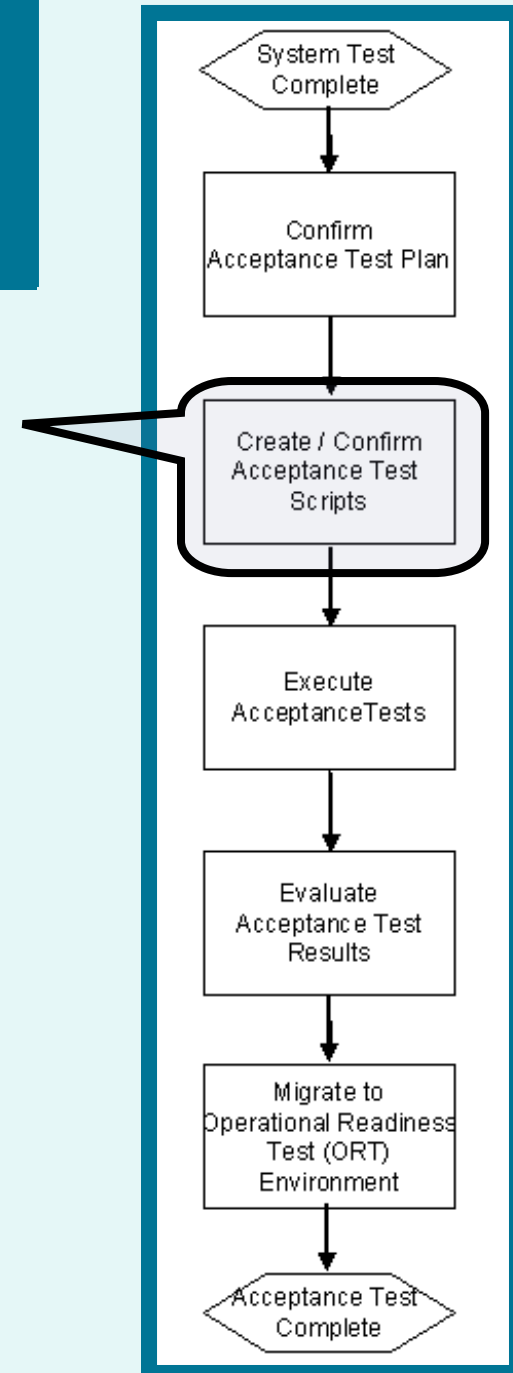
## Create/Confirm Acceptance Test Scripts

If acceptance test scripts were created during an earlier sub-phase, the correctness of the scripts can simply be confirmed.

If acceptance test scripts were not created earlier, they should be created from test cases and scenarios prior to the initiation of actual acceptance testing.

Test scripts should document all test conditions, data, and expected results.

The completeness and correctness of the acceptance test scripts should be validated against the updated requirements traceability matrix.

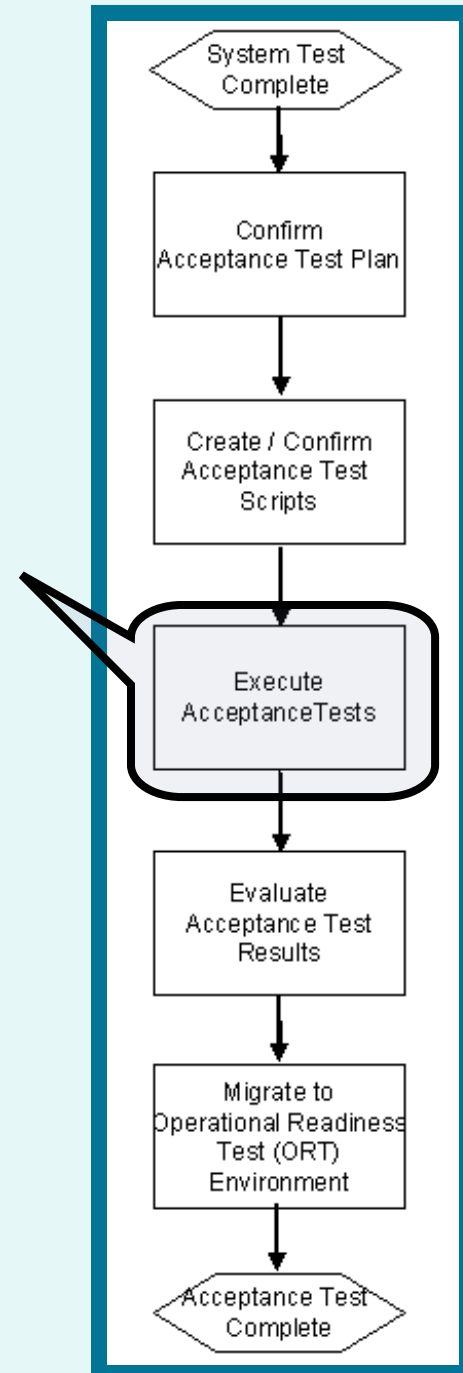


## Perform Acceptance Test

# Execute Acceptance Tests

Successful acceptance testing demonstrates that the application meets the business community's requirements from the following perspectives:

- Functionality
- Regression
- Usability
- Business process
- Integration.



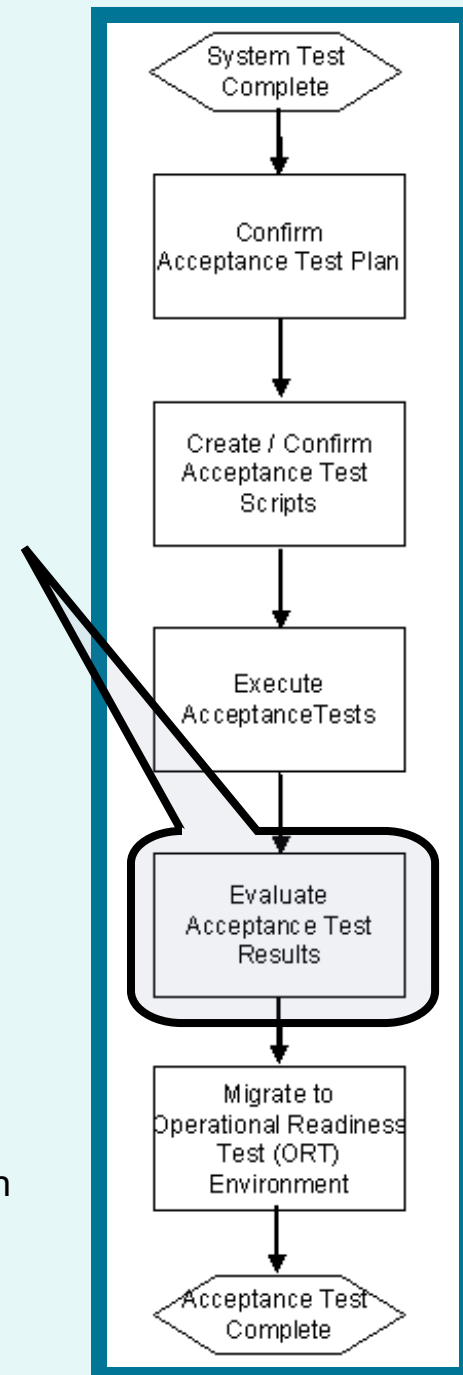
## Perform Acceptance Test

## Evaluate Acceptance Test Results

The Evaluate Test sub-process is initiated once the Execute Test sub-process is complete to determine if the test was successful. A successful test allows the application or solution iteration to move to the next test phase. An unsuccessful test requires errors to be fixed and the test re-executed. After the last phase of testing complete (i.e. Operational Readiness Testing), an Operations Certification is rendered to indicate whether the application or product is ready to move to production.

The Evaluate Test sub-process consists of the following steps:

- Compile and document test results.
- Evaluate the success of the test.
- Log defects for tracking purposes.
- Determine if the application/solution iteration passed or failed the test. Consider the acceptance criteria defined in the test plan for this test phase. For a failure, the reason for the failure must be determined (e.g. script error, software error).
- Identify opportunities to improve the testing process.
- Complete testing Quality Gates assessments.
- Initiate version control or change management, if application/solution iteration is ready for production.



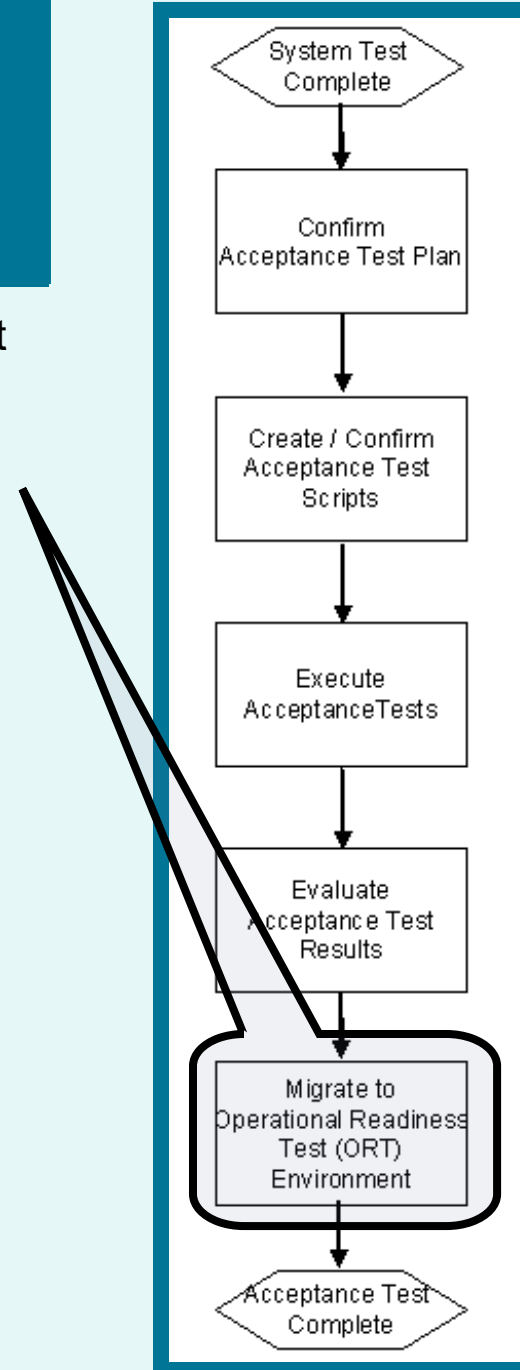


**Perform  
Acceptance  
Test**

## Migrate to Operational Readiness Test (ORT) Environment

Migration from the acceptance test environment to the operational readiness test environment involves two major tasks:

- First, the environment is prepared for the migration. Since migrations between the acceptance and operational readiness test environments may occur iteratively, each iteration must start with preparation/confirmation that the operational readiness test environment contains the correct version of all necessary elements, no extraneous elements, the correct version of all necessary data and no extraneous data.
- 
- Second, only then should new or updated solution elements be migrated into the operational readiness test environment.



**Perform  
Acceptance  
Test**

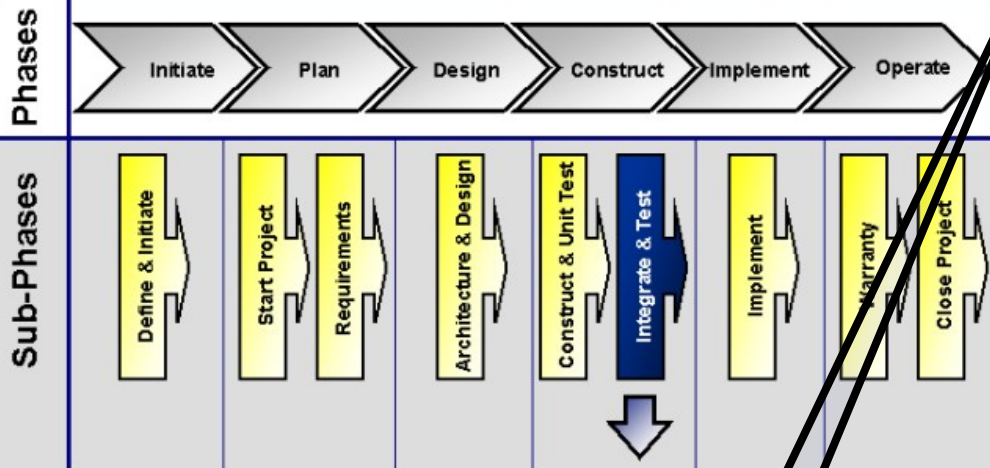
# Migrate to Operational Readiness Test (ORT) Environment

<b>Inputs &amp; Outputs</b>	
<b>Inputs</b>	Integrated Solution
<b>Outputs</b>	Migrated Solution
<b>Roles &amp; Responsibilities</b>	
<b>Role</b>	<b>Responsibility</b>
Process and Application Team Leader	Executes this activity. Confirms that the state of the operational readiness test environment is appropriate for migration. Performs migration.
Process and Application Team	Supports this activity. Assists in confirming that all necessary application elements exist in the operational readiness test environment.
Data Design & Integration Team	Supports this activity. Assists in confirming that all necessary data elements exist in the operational readiness test environment.
Infrastructure Engineer	Supports this activity. Assists in confirming that the environment is correctly configured.



# Detail - Perform Operational Readiness Test

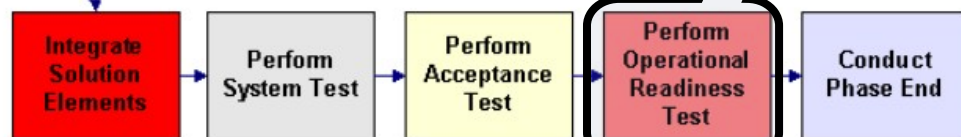
## Phases and Sub-Phases of the Project Lifecycle



6

Integrate & Test

## Integrate and Test Sub-Phase



Successful operational readiness testing demonstrates that the application meets operational and configuration requirements, including:

- Requirements for recovery
- Day-to-day operations
- Resource usage
- Configuration
- Interoperability
- Security
- Installation and backout.



# Perform Operational Readiness Test

## Inputs & Outputs

<b>Inputs</b>	Requirements Specification Requirements Management Deliverables, including an up-to-date Requirements Traceability Matrix Detailed Unit Testing Deliverables System Implementation Deliverables
<b>Outputs</b>	Operational Readiness Test Results Operational Readiness Test Evaluation Defect Log

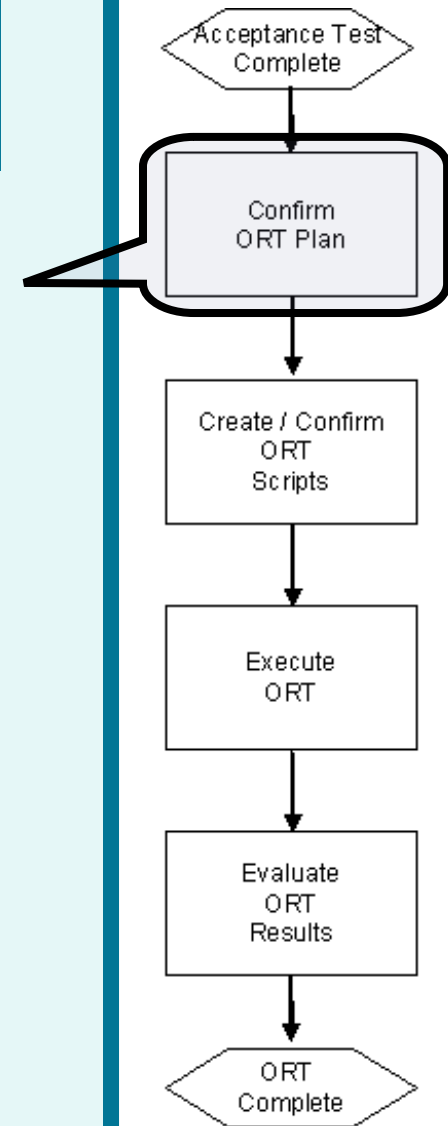
This process is broken out into 4 sub-processes:

- Confirm Operational Readiness Test Plan
- Create/Confirm Operational Readiness Test Scripts
- Execute Operational Readiness Test
- Evaluate Operational Readiness Test Results

**Perform  
Operational  
Readiness  
Test**

## Confirm ORT Plan

The operational readiness test plan is created during the Architecture and Design Sub-Phase and is confirmed at the beginning of operational readiness test phase.



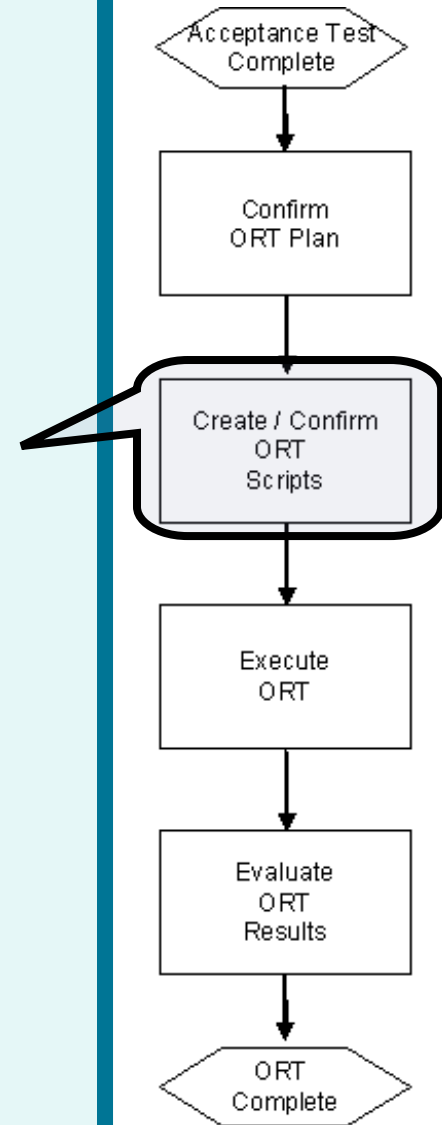
**Perform  
Operational  
Readiness  
Test**

## Create/Confirm ORT Scripts

If operational readiness test scripts were created during an earlier sub-phase, the correctness of the scripts can simply be confirmed.

If operational readiness test scripts were not created earlier, they should be created from test cases and scenarios prior to the initiation of actual operational readiness testing.

Test scripts should document all test conditions, data, and expected results.



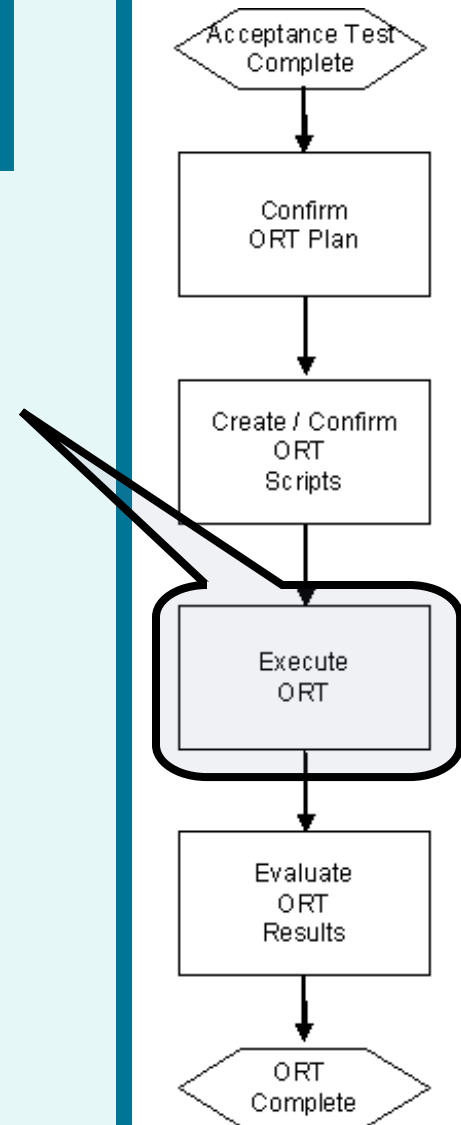
**Perform  
Operational  
Readiness  
Test**

## Execute ORT

Successful operational readiness testing demonstrates that the application meets operational and configuration requirements, including:

- Requirements for Recovery
- Day-to-day Operations
- Resource Usage
- Configuration
- Interoperability
- Security
- Installation and Backout.

Upon successful completion of operational readiness testing, the application is ready for implementation.



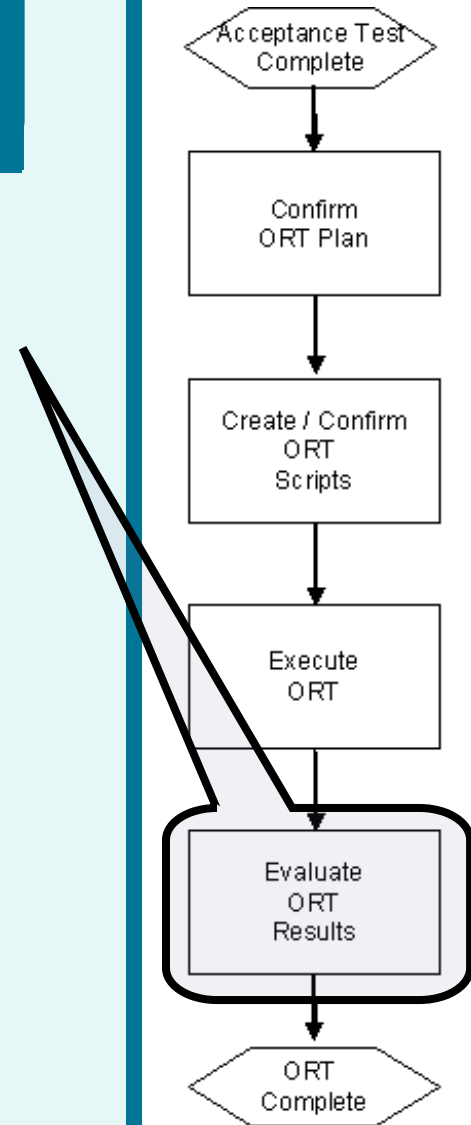
## Perform Operational Readiness Test

# Evaluate ORT Results

The Evaluate Test sub-process is initiated once the Execute Test sub-process is complete to determine if the test was successful. A successful test allows the application or solution iteration to move to the next test phase. An unsuccessful test requires errors to be fixed and the test re-executed. After the last phase of testing complete (i.e. Operational Readiness Testing), an Operations Certification is rendered to indicate whether the application or product is ready to move to production.

The Evaluate Test sub-process consists of the following steps:

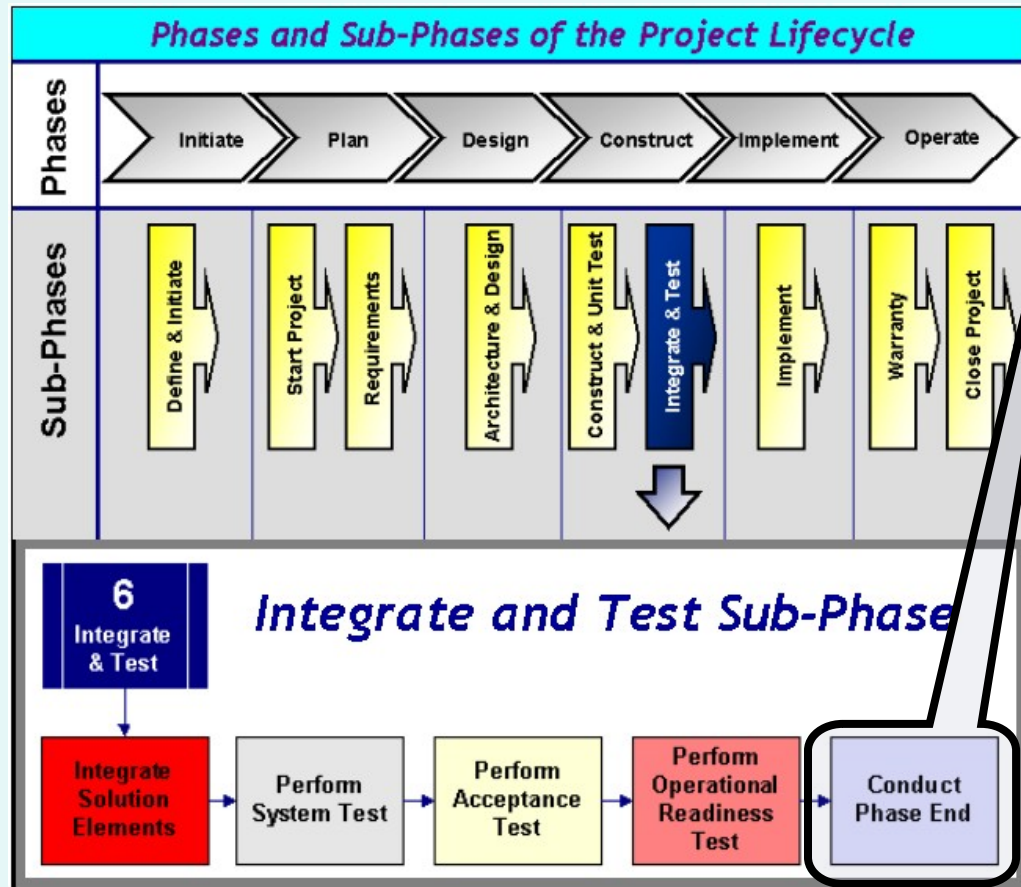
- Compile and document test results.
- Evaluate the success of the test.
- Log defects for tracking purposes.
- Determine if the application/solution iteration passed or failed the test. Consider the acceptance criteria defined in the test plan for this test phase. For a failure, the reason for the failure must be determined (e.g. script error, software error).
- Identify opportunities to improve the testing process.
- Complete testing Quality Gates assessments.
- Initiate version control or change management, if application/solution iteration is ready for production.



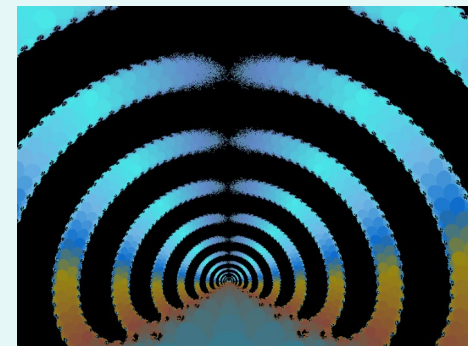




# Detail - Conduct Phase End



The standard project phases, as defined in the project lifecycle, incorporate phase end gates or "decision points" to allow for a senior management review. Just as a Business Case (with Cost Benefit Analysis) is presented to senior management for funding in the Define & Initiate sub-phase, additional validations of the project are performed at the subsequent phase end gates (e.g. Plan, Design, Construct).





# Conduct Phase End

To prepare for this review, a Project Manager would ensure the quality of their key deliverables produced during the sub-phase (see Quality Management), confirm the required Information Protection deliverables, update their project workplan for the next phase (see Planning & Estimating) and re-calculate the project costs and benefits based on progress to date, comparing to those originally approved in the Define & Initiate Sub-Phase.

These phase end gates or "decision points" enable Project Managers and leaders to the review and evaluate the project to ensure continued alignment with business objectives; review project financial and deliverables progress, and provide guidance and funding for the remaining phases.

After reviewing the results of the testing effort, if it is determined by management that at this point the project is ready to proceed, the project will move to the Implementation Phase. If management deems further testing/fixing needs to be done, the project team will perform the requested additional testing/fixing. Once the requested additional testing effort is finished, the project team will inform management and proceed to the Implementation Phase.



# Conduct Phase End

<b>Inputs &amp; Outputs</b>	
<b>Inputs</b>	All Phase deliverables Business Case Project Charter Project Workplan
<b>Outputs</b>	Test Results Updated Business Case Updated Project Workplan “Go” or “No Go” project decision
<b>Roles &amp; Responsibilities</b>	
<b>Role</b>	<b>Responsibility</b>
Project Manager	Produce a final document that captures all the test results for the proposed system. Update Business Case, Project Charter and workplan. Prepare documentation and a presentation of project status, and present at Phase End Review.
Project Sponsor	Provide guidance to the Project Manager in preparation for Phase End Review.
Steering Committee	Participant in Phase End Review, provide direction when necessary, and provide a “go” or “no go” decision for project to enter the next phase.



# Integrate & Test Summary

## ***Purpose***

- **The purpose of the Integrate & Test Sub-Phase is to assemble all application functionality into a coherent whole and test the constructed functionality.**

## ***Team Roles***

- **Steering Committee, Project Sponsor, Project Manager, Support Areas, Project Office, Process and Application Team, Testing Team, Technical and Infrastructure Team, Data Design and Integration Team, Organization Design and Enablement Team, Training and Documentation Team, Implementation Team.**

## ***Major Inputs***

- **Requirements Management Deliverables**
- **Detailed Unit Testing Deliverables**
- **System Implementation Deliverables**
- **Requirements Specification**

## ***Major Outputs***

- **(Updated) Requirements Management Deliverables**
- **Detailed Acceptance Testing Deliverables**
- **Detailed System Testing Deliverables**
- **Detailed Operational Readiness Test Deliverables**
- **Fully Integrated Solution**

## ***Processes***

- **Integrate Solution Elements**
- **Perform System Test**
- **Perform Acceptance Test**
- **Perform Operational Readiness Test**
- **Conduct Phase End**



# Integrate & Test Sub-Phase

*Thank You*

