

EXHIBIT L

Application Development Processes

Optum Development Methodology

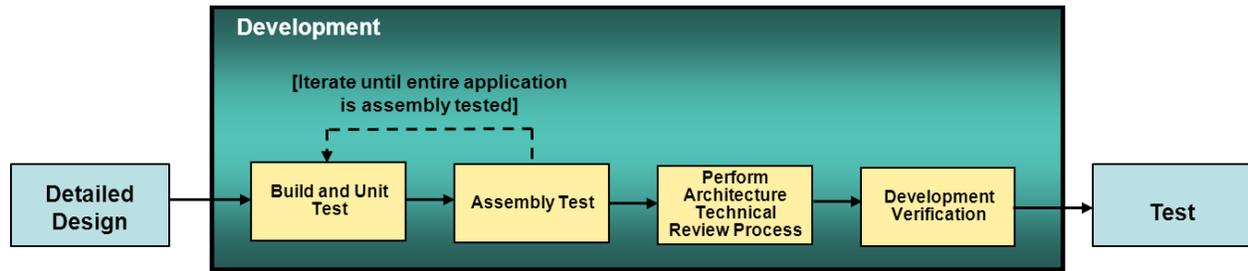
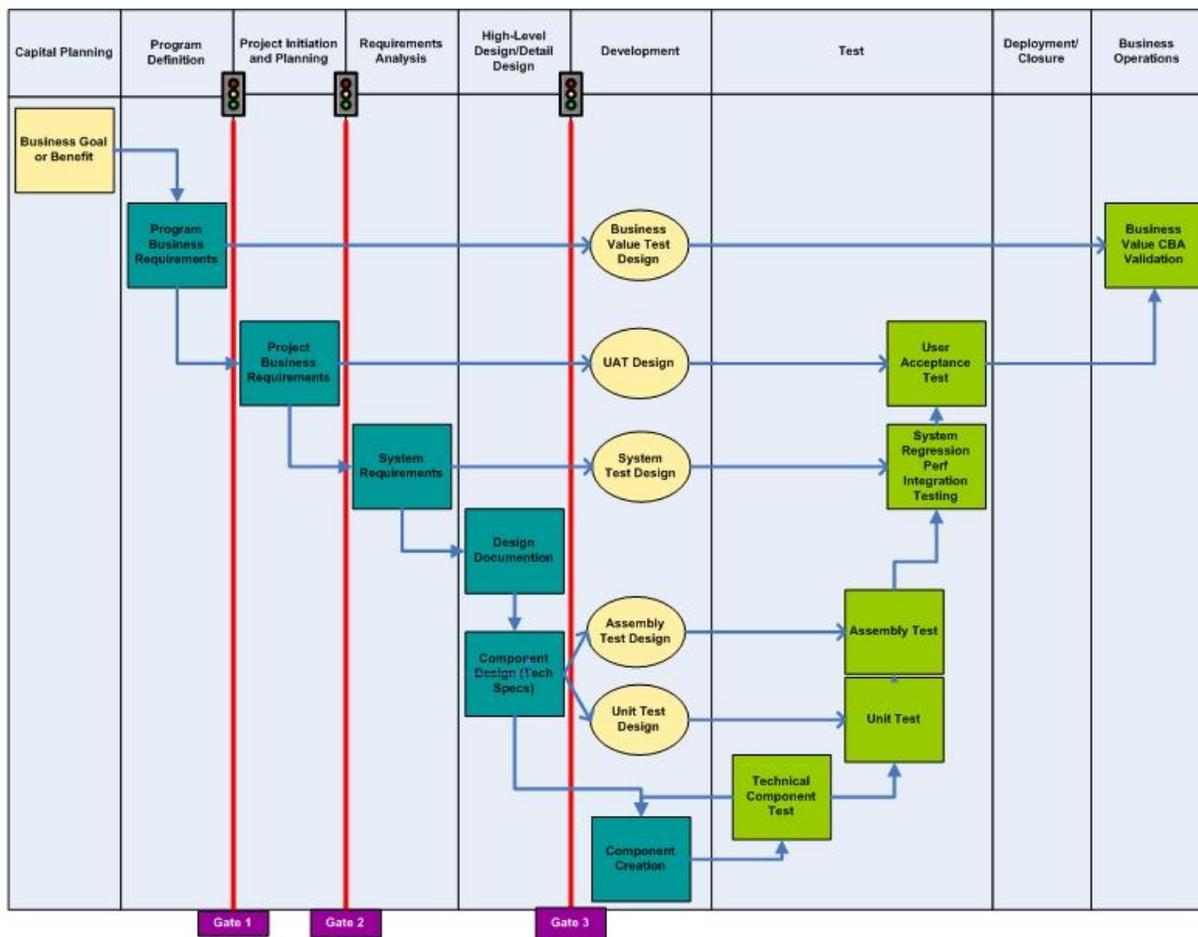


Figure 1: Development process flow

Development Overview

The Development phase consists of activities that include the building, code review, unit testing and assembly testing of all release components (i.e. user interfaces, program code, interfaces, databases, etc.). Prior to the start of System Test execution, development verification is conducted to ensure all activities have been completed successfully.

The Optum Development Methodology (ODM) defines various testing activities. The Test Model diagram below depicts how each testing activity ties back to the various phases within ODM.



Detailed Design and Build Activities

The left side of the Development model demonstrates the design activities completed during the release lifecycle. This includes:

- Requirements analysis - identifies application specifications
- High-level design - identifies architectural components to meet requirements
- Detailed design - identifies detailed behavior of components

Build

Application build constructs the required components. This includes:

Database Components

Database Component Build follows the design activities for the database from the Data Architecture Definition and Data Architecture Design activities. In this step, the approved designs are executed in the development environment for the release. This includes new database objects, modifications to existing databases, and interim data components specified in the designs. This step applies to any type of database regardless of platform.

This step may not be necessary if there is no database work required for the release. However, when database work is part of the release, this step **must** happen prior to the building of Application components that are dependent upon the database implementation. For application components that are not dependent on database changes, the build activities may proceed in parallel.

To build a database component, the following activities are performed:

- Obtain the appropriate Physical Data Model, Data Conversion Design, Application Interface Specifications artifacts for the release. Utilize the data architect to gain knowledge about the details.
- Review the artifacts with the development lead and/or developer and follow-up on any outstanding questions.
- Perform the relevant activities in the database as described in the detailed design artifacts.
- Review the component being developed with release team members and design partners to ensure that the component meets the design and application specifications. Ensure that the component is synchronized with other build activities.

Guidelines (required):

- Use of automation and data modeling tools aid in maintainability of the database.
- Use of version control and base lining of all code.
- Maintain and update documented code via the standards document included in the reference section below.

Application Components

Utilizing the Technical Specifications and Data Conversion Design artifacts developed within the Detailed Design phase, create the components, including programs, screens, Web objects, batch

programs, reports, interfaces, etc. All components must follow the application development standards specified for the type of component being built.

To build an application component, conduct the following activities:

- Obtain the appropriate Technical Specifications and/or Data Conversion Design, Application Interface Specification Document artifact(s) for the release.
- Review the artifacts with the Development Lead and team and resolve any outstanding questions or issues.
- Perform the relevant development activities in the application as described in the artifacts.
- Check out source code from appropriate version control system. Development teams must ensure that source code is protected per Optum corporate Security Policies.
- Perform code changes to the component, following Optum's documented application coding standards.
- Changes to batch components are required to adhere to the Optum Batch Standards. The Batch Standards define how to handle files, reports, error processing, restarts and job and file naming conventions.
- Add comments to the code explaining why the changes are being made, per Optum documentation standards.
- Perform peer review of code per Optum peer review standards.
- Compile code to ensure a clean compile and build.
- Use code scanning tool output as appropriate (e.g., checkstyle, PMD, FindBugs, etc.).
- Ensure that the component is synchronized with other build activities.

Guidelines (required):

- When building or modifying Application interfaces, use the Application Interface Specifications document.
- When organizing the sequence of Application components being built, consider completing work on Data Conversion(s) first, since Unit Testing of other components may depend on data converted by these programs.
- Use appropriate development tools approved by Optum during development.
- Use version control and Baseline all code.
- Maintain and update documented code via the standards document included in the reference section below.
- Store artifacts in the application's document management.

Code Review

Applications are required to complete code reviews for all projects.

The primary purpose of a code review is to critique the code and remove coding defects during the development phase lifecycle, prior to promoting the code to System Test.

Code reviews are conducted after a successful build. Teams have the option of reviewing code by author, project, or module/component.

Activities required to complete the code review process are:

- The Development Lead (or delegate) identifies the reviewers and schedules the review meeting

- Prior to the meeting, the reviewers conduct an independent review of the code and complete the Code Review and Defect Tracking Workbook.
- During the review meeting, the reviewers walk through their findings with the Author.
- The Development Lead updates the Code Review and Defect Tracking Workbook with the agreed upon defects.
- The Author performs all required rework.

Note: If updates to the code are substantial, a follow-up peer review is conducted.

- The Author updates the Code Review and Defect Tracking Workbook with the defect resolution and final status and stores the artifact in the application's document management tool.

Guidelines:

- Reference ODM code review procedures for step by step instructions on completing a code review.
- Utilize all available code scanning tools to uncover and correct all defects prior to conducting the code review.
- Application Development must follow the Release Entry Framework (REF) Process and include Production Operational Support in the code review meetings.
- Release artifacts must be available for audit **36** months post production.
- Follow Enterprise Records Management Retention Policy for document management thereafter.

Unit Test

Unit Test verifies that individual units of source code are working properly. The objective of unit testing is to test not only the functionality of the code, but to ensure that the code is structurally sound and robust and able to respond appropriately in all conditions. Unit test includes the following activities:

- Creation of unit test scripts
- Execution of unit test scripts
- Validation of unit test scripts

It is a recommended best practice that the following entry criteria be met before proceeding with Unit Test:

- The Test Plan has been approved by the required stakeholders
- Each component has been coded and successfully compiled
- Code has undergone code review
- Unit Test environment is created and available

Guidelines (required):

- Reference High Level Design and/or Data Conversion designs, to create test scripts for the specific development component.
- Execute Unit Test Scripts. Repeat, as necessary, if fixes are required to the component. Execute until all test conditions have been executed successfully.

- All IT Application teams are required to track and manage software defects. If an automated Tool for Defect Tracking and Management is used, refer to tool usage procedures or Best Practices documentation. Defect tracking is critical for analyzing defect trends and proposing process improvements which focus on defect prevention.
- Consider the use of tools to automate unit testing where possible
- If the release includes batch applications, refer to the Optum Batch Standards for required batch standards.
- Ensure that Unit Test considers all technical specifications for each component, including access control, Applications security features and Design Patterns. Example Test Criteria for information security threat countermeasures are available in the ODM Artifact Repository.
- For Web Applications, refer to the Web Deployment Checklist sheet of the Deployment Readiness Checklist for additional security activities to be started at this time.
- Upon completion of Unit Test, ensure all debug settings and logins that are not needed for User Acceptance Test have been disabled. Provide a list of those settings that have not been disabled to the Release Manager for inclusion in the Deployment Readiness Checklist.

Assembly Test

Assembly Test ensures that related components function properly when assembled together. Testing the application component interfaces verifies that correct data is passed between components. At the completion of Assembly Testing, all component interfaces within the Application are executed and verified that they are working according to specifications. Assembly Test follows Unit Test and precedes System Test.

It is a recommended best practice that the following entry criteria be met before proceeding with Assembly Test:

- All components within the application have been Unit Tested
- Identified defects in a given component have been corrected and re-tested or documented with the reason the defect has not been fixed.
- Assembly Test environment is created and available

Guidelines:

- Ensure that Assembly Test includes all impacted components.
- Reference High Level Design documents when creating Assembly Test Scripts
- Track and manage software defects. If an automated Tool for Defect Tracking and Management is used, refer to tool usage procedures or Best Practices documentation. Defect tracking is critical for analyzing defect trends and proposing process improvements which focus on defect prevention.
- Upon completion of Assembly-level testing, all defects that remain unresolved are documented and communicated to the Test Organization during Entry Criteria Review Session. It is a System Test Entry Criteria Requirement to review all open/outstanding Assembly Test defects in order to determine that they are not showstoppers for System test.
- Final Assembly test results should be documented via the Test Tool or in the Test Summary Report.
- Upon completion of Assembly Test, ensure all test code is removed from the Application build before System Test.