

Testing Window Concepts:

- Deliveries to test are scheduled for every 20 work days. All promotions or changes to the test environment are scheduled around these delivery dates, including:
 - New development task orders
 - Maintenance tasks
 - J2EE Conversion deliveries
 - Operational Environment upgrades and configuration changes
- Each test delivery equates to an iteration. An iteration may move forward to acceptance testing and production or may be only an integration test iteration.
- Iterations should be sufficiently granular so that they can be integration tested within the 20 day integration test period, allowing for at least one bug fix release within this period.
- For iterations that are proceeding to production, the integration testing window will be followed by a 10 work day acceptance testing period.
- If Development misses a delivery for a testing window, the delivery will be pushed back to the next testing window.
- ClearQuest will be modified to support this concept. Instead of tracking individual test and production dates for each task, each task will simply be associated with a testing window which will have a fixed set of dates.
- Under special circumstance, it is possible to identify a delivery as being 'fast tracked'. This delivery will not follow the normal structure and will be tested and deployed as soon as possible. This is highly discouraged except for extreme cases since these types of releases are very difficult to manage. Putting a delivery through a 'fast track' will require special CCB approval.

Testing Window Advantages:

- Improved manageability for testing activities. Currently the ad-hoc deliveries to test make it very difficult to manage testing resources and activities.
- Improved planning for development activities. Having a fixed plan for initial and bug fix releases simplifies the management of development resources and helps ensure that development resources are available to quickly address critical defects during the testing period.
- Improved stability of testing environment. Currently with almost daily releases to integration test, the environment is not stable or consistent.
- Improved consistency with production environment. Currently since release go to production "when they're ready" rather than with all other releases that they went through the test cycle with, the production environment they are deployed into may not be consistent with the test environment they were tested in.
- Improved accountability for development activities. Based on the testing window, development can be more easily held accountable for on-time delivery to testing as well as the quality of the delivery (if an iteration contains critical defects that cannot be fixed within the integration testing period it would be considered a failed delivery).
- Improved accountability for test activities. Testing groups can be held accountable for meeting the schedules of each testing window.

Detailed schedule for 20 work day integration test and build verification period:

- Day -2: Prior to the start of the cycle and the completion of unit testing, a Deployment Readiness Review will be held to ensure that all information has been correctly specified for the upcoming releases to test. This will require the CQ tickets to be filled out with the relevant build information.
- Day 1: Build execution and environment setup.
- Day 2-3: Verification (smoke test) of environment. This will be performed by developers and testers together to ensure that the expected build was correctly delivered and deployed into the test environment. Any build/deployment issues discovered during this time will be corrected by Development, CM, and Ops working together. At the end of Day 3 a Test Readiness Review (TRR) is held to ensure that all applications are ready to start full integration testing.
- Days 4-11: Completion of first round of testing (Test Cycle 1), with complete coverage of new or changed functionality Task Order Managers review defects found and assign defect correction task to development with list of defects to be fixed. This will be an ongoing activity throughout the test cycle.
- Day 12: Reserved for development to address any critical defects discovered in the final day of Test Cycle 1 and unit test and deliver defect correction release.
- Day 13: Defect correction build execution, environment setup and verification.
- Day 14-20: Testers verify defects (Test Cycle 2). Testers also perform full regression testing once there are no outstanding critical defects. If critical defects are found, it may be possible for developers and testers to turn around another defect correction release within this period. Task Order Managers will need to coordinate these on a case-by-case basis. If critical defects are found that can not be corrected and verified within this test cycle, the application is pulled from this release and scheduled to go into a future release cycle (note that this will require a CCB as it may affect other activities planned for that cycle).

Detailed schedule for 10 work day acceptance test period:

- Days 1-2: Setup and verification of environment
- Days 3-10: Completion of acceptance testing. If any critical defects are found, the application is pulled from this release and schedule to go into a future release cycle (note that this will require a CCB as it may affect other activities planned for that cycle).

The following graph is an illustration of what the new testing window will look like:

Iteration 1

Cm Build Release to SIT and Environment Validation (3 Days)

Test Cycle 1 (8 Days) – Initial complete test run

CM New Build to fix Critical Defects (1 Day)

Test Cycle 2 (7 Days) – Defect Fix and regression test run