



Quality Program Office Fit for Promote Initiative

Ensuring project quality before handoff

Fit for promote

- Goal:
- Ensure project quality before handoff from development to product validation
 - 100% PV FST pass rate and 80% or over pass rate for other tests
- Review focuses on:
 - Testing and validation before every project promote
 - Accountability of development teams and leads for the quality of their project deliverables

Fit For Promote

- Required gate for ALL project promotes to the release node
- CPs for projects that have not completed FFP will be rejected
- Schedule reviews early, do not leave until the last minute
- Interim Promotes (non-PV handoff promotes)
 - Lightweight subset of the full checklist
 - Focuses on maintaining baseline stability and preventing regressions
 - Can typically be completed in 30 minutes
- Points of emphasis
 - Code Reviews – the code is reviewed, complete, and ready
 - Auto tests, integration, regression – the code is stable, won't break the build or cause regressions
 - Platform coverage – the code builds and runs

Fit for promote

- Full Promotes
 - Focuses on project completeness, readiness, robustness
 - Typically 60 – 90 minutes depending on size of project
 - Points of emphasis
 - Code Reviews – the code is reviewed, complete, and ready
 - Unit tests, integration, regression – the code is stable, won't break the build or cause regressions
 - Platform coverage – the code builds and runs
 - Project testing – complete, thorough, clean
 - Upgrade testing – upgrade has been tested and works
 - Memory, performance, scalability – has been tested, results analyzed
- N/A answers are used too frequently
 - Rarely is this an acceptable answer
 - You must include a detailed justification for any N/A answer
- Look at test results of all kinds – be prepared

Fit for promote - Attendees

- Project owner – The tech lead or the manager responsible for the project. Will present the project status and clarifications
- Quality program office representative -- Provides final assessment and disposition of the outcome
- Product Validation representative – Assess if PV has any concerns regarding the quality of the project promote (optional if non PV handoff)
- Program Management – If the project is under a program auspices (optional based on project)
- Release Management representative – Assess promote timeframe to ensure that multiple projects do not go in the same build (optional based on project)
- Invited:
 - Domain manager
 - Product management
 - Design Office

Fit for promote meeting

- Before the meeting:
 - Project Owner
 - Schedules the meeting
 - Prepare checklist and all associated data
 - Test cases
 - Test results
 - Code review results
 - Check-in checklist into SDLC
 - Submitted change package number
 - After the meeting:
 - Project Owner: update checklist with result in SDLC
 - Quality Office Reviewer: enter Fit for Quality Status, log Fit for Quality Issues
 - Meeting Agenda:
 - 5 to 10 minutes – Project functionality and value to customer & what is included in this promote
 - 50 to 60 minutes – Walk down checklist, code review results, show tests and results, etc.

Fit for promote – tracking and disposition

- Fit for promote results will be tracked at the release sites
- Fit for promote disposition
 - Red – Not fit for promote
 - Action items and reschedule meeting
 - Yellow – Not fit for promote
 - Action items to be followed up via. email and update in checklist
 - Date for build based on build and promote schedule
 - Green – Fit for promote
 - Action items to be followed up via. email and update in checklist
 - Date for final build based on build and promote schedule

Fit for promote – Areas of focus

- Negative testing – This is where we find the largest number of issues
- Impacts / dependencies to other domains
- Quality of testing
 - Unit tests, PTP tests and integration tests
 - Thoroughness of tests
 - More automated & persistent, the better
 - Number of tests (coverage)
 - Platform coverage
- Failures – what areas are weak and why?
- Exceptions are generally not made for test case failure
 - Happy path (FST / PTP) must work
 - No crash / hangs
 - Negative input should not cause bad behavior
 - Auto and Xunit tests must pass successfully
 - Regressions in other areas caused by project code
- No memory leaks
- Performance tests
 - Compare with before state
 - Auto test performance and SQL counts
 - Scalability (does the code scale with size of data)
- Sufficient code review? Was there adherence to “new” ways to implement?

Fit for promote focus area

The need to improve code reviews

- Quality Program objectives based on review and feedback
- Peer Code Reviews are a valuable tool to:
 - Identify issues prior to PV hand-off
 - Encourage collaboration
 - Spread knowledge and increase developers' areas of competence
 - Produce code that is more maintainable
- Peer Code Reviews are **not being done consistently and effectively** across all projects and all domain areas

- We are **missing an opportunity to catch and fix issues** earlier in the process where they are **less expensive**

Fit for promote Code review objectives

- Objectives:
 - To ensure that quality code reviews were conducted for the project
 - To ensure that issues/observations were tracked and corrected
 - To ensure that code reviews are consistently and effectively applied to all projects in all domains, share best practices
- Wiki site for Code Reviews has been started
 - Share best practices
 - Share information, links to resources
 - Provide examples
- Authors have editors, Musicians have producers ... Software developers have code reviews

Fit For Promote – Code review process

- Applies to ALL project promotes
 - The checklist has been updated
- What are we looking for?
 - Code review methodology used in the project
 - Quality of the reviewers and the complexity of the change
 - Detailed review of the results
 - Recording, tracking, and closure of all code review issues
- Do
 - Ensure your project is doing frequent, thorough, quality code reviews
 - Ensure the results are recorded, issues logged, and tracked
 - Resolve the issues identified in your code reviews
 - Record your time in Timecard appropriately
- Don't
 - Wait until the last minute to conduct code reviews

Thank you