Managing Workflow for Test Case Management

July 2006 - Pragmatic Software Newsletters

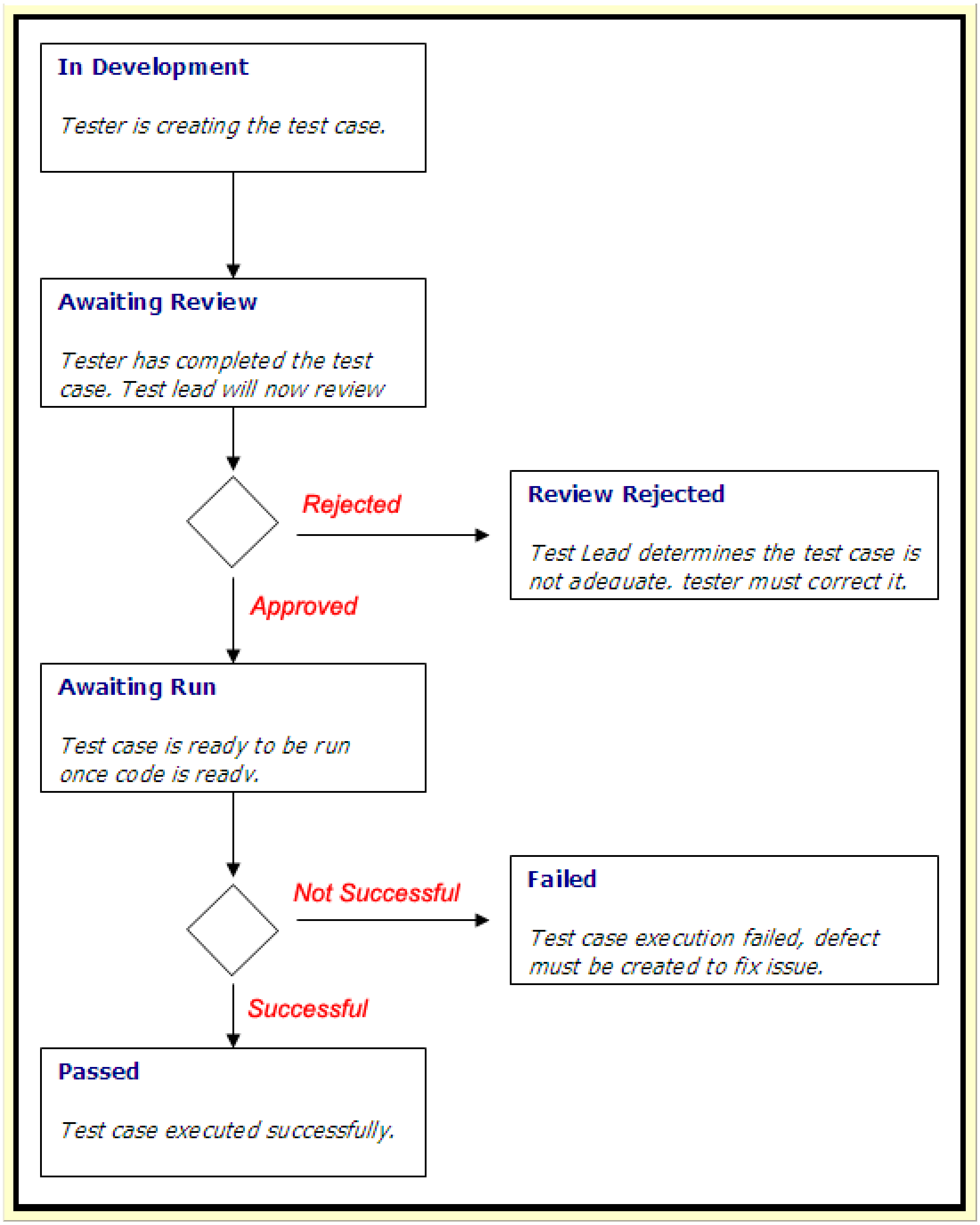
For teams managing software quality, it is crucial to define solid test cases for each customer requirement. Just as importantly, the workflow around the test case management process must be defined so that everyone understands how a test case moves from definition to a point where it is ready for execution. Below are some tips for defining test cases and the workflow.

1. **Define the Workflow Statuses** - When developing test cases, it is important to define the workflow. For larger teams that have a test lead, the test lead should provide oversight and approval for all test cases, to ensure depth and robustness of the test suite. For smaller teams, this approval process for test cases could be eliminated (assuming the fewer number of testers have a high skill level). Workflow is normally tracked via the "status". Let's create a simple workflow for a larger team, where the tester creates a test case, the test lead approves the test case, and the tester executes the test case once testing begins. Below are some possible sets of statuses (workflow) for this process.

   Workflow Statuses:

   In Development
   Awaiting Review
   Review Rejected
   Awaiting Run
   Passed
   Failed

2. **Flowchart the Workflow** - Flowcharting the workflow allows team members to understand the process in full. We created the flowchart below using Microsoft Word.



In Development

Tester is creating the test case.

Awaiting Review

Tester has completed the test case. Test lead will now review

Review Rejected

Test Lead determines the test case is not adequate. tester must correct it.

Awaiting Run

Test case is ready to be run once code is readv.

Failed

Test case execution failed, defect must be created to fix issue.

Passed

Test case executed successfully.

3. **Advanced Workflow** - In our example above, we used simple workflow. However, if your team uses software to manage customer requirements, you should be able to implement more robust workflow. For example, the software should allow you to define "state transitions". This identifies how a status can transition from one status to another. In our example, above, you may want to only transition from **Awaiting Review** to either **Review Rejected** or **Awaiting Run**, because none of the other statuses would be applicable at that point in your workflow. Likewise, the software should also allow you to define what fields (or items) you wish to make required upon different states. In the example above, if the requirement is changed to **Awaiting Review**, we may want to require that the tester enter the **Test Case Steps** and **Test Case Expected Results**. Robust test case tracking software will allow you to define the field attributes for each state transition. **Software Planner** (http://www.SoftwarePlanner.com) does this nicely, you can see how this is handled from **Software Planner** by viewing this movie:

http://www.pragmaticsw.com/GuidedTours/Default.asp?FileName=Workflow

4. **Test Case Traceability** - As you develop your test cases, you should ensure that you have an adequate number of test cases for each customer requirement. This ensures that you have adequate testing for each customer requirement. How do you do this?

   Make a list of every test case you have defined, and link each test case back to the requirement. Then meet with your project team (project manager, developers and testers) to review the test cases. This meeting will spawn new test cases you may not have thought of and will allow everyone visibility into your test process.

   Basic Approach - If you do not have a software development lifecycle tool, a low-cost and simple approach to this is to create a spreadsheet that contains a list of your functional specifications (requirements) and lists each test case that has been created for each requirement. **Example:** TestCase_Traceability.xls

   Advanced Approach - A better approach is to utilize an SDLC tool that allows tracking of functional specifications (customer requirements), and allows creating of test cases that are linked to the functional specification. The software should provide a traceability report that shows how the test cases are linked to each requirement. Below are some example reports that come bundled with Software Planner, but any good SDLC tool should provide these types of reports.

   Example: Software Planner_TestCase_Traceability Report.pdf Tracking Requirements Tracking Test Cases

5. **Test Plan Template** - It is also wise to have a detailed test plan so that you are sure that your test lead and project management agrees with your approach and test design. Here is a test plan template:

   http://www.PragmaticSW.com/Pragmatic/Templates/TestDesign.rtf

Helpful Templates

Below are some helpful templates to aid you in developing software solutions on-time and on-budget:

- **Project Management Guidelines -** http://www.PragmaticSW.com/Pragmatic/Templates/ProjectMgtGuidelines.rtf
- **Functional Specifications -** http://www.PragmaticSW.com/Pragmatic/Templates/FunctionalSpec.rtf
- **Architectural Overview -** http://www.PragmaticSW.com/Pragmatic/Templates/ArchitectureOverview.rtf
- **Detailed Design** - http://www.PragmaticSW.com/Pragmatic/Templates/DetailedDesign.rtf
- **Strategic Planning Document** - http://www.PragmaticSW.com/Pragmatic/Templates/StrategicPlanning.rtf
- **Test Design** - http://www.PragmaticSW.com/Pragmatic/Templates/TestDesign.rtf
- **Risk Assessment** - http://www.PragmaticSW.com/Pragmatic/Templates/Risk%20Assessment.rtf
- **Weekly Status** - http://www.PragmaticSW.com/Pragmatic/Templates/WeeklyStatusRpt.rtf
- **User Acceptance Test Release Report** - http://www.PragmaticSW.com/Pragmatic/Templates/UATRelease.rtf
- **Post Mortem Report** - http://www.PragmaticSW.com/Pragmatic/Templates/PostMortem.rtf
- **All Templates** - http://www.PragmaticSW.com/Templates.htm
- **Prior Newsletters** - http://www.PragmaticSW.com/Newsletters.htm
- **Software Planner** - http://www.SoftwarePlanner.com
- **Defect Tracker** - http://www.DefectTracker.com
- **Remoteus (Remote Desktop Sharing)** - http://www.PragmaticSW.com/Remoteus.asp

About the Author

Steve Miller is the President of **Pragmatic Software** (http://www.PragmaticSW.com). With over 20 years of experience, Steve has extensive knowledge in project management, software architecture and test design. Steve publishes a monthly newsletter for companies that design and develop software. You can read other newsletters at http://www.PragmaticSW.com/Newsletters.htm. Steve's email is steve.miller@PragmaticSW.com.

Pragmatic Software Co., Inc.
383 Inverness Parkway
Suite 280
Englewood, CO 80112

Phone: 303.768.7480
Fax: 303.768.7481
Web site: http://www.PragmaticSW.com
E-mail: info@PragmaticSW.com