deliver early, deliver often

RALLY
software development

Using Rally to Support Software Documentation

1655 Walnut Street, Suite 200
Boulder, Colorado 80302

v 303 226 1180
f 303 226 1179
www.rallydev.com

Introduction

When we document a new or updated functionality for a Rally release, we employ many of the same practices used in good Agile development. Changing requirements are probably the biggest challenge facing the documenter in an Agile environment and using Rally allows us to embrace any change that occurs. If we can keep a handle on the changes that are being implemented, we can remain in step with the developers as they work their way through the Story Cards and Tasks of an Iteration.

How *Rally* Uses Rally

The following steps outline our basic process at Rally for documenting functionality of a Release when the artifacts are kept in Rally's software.

1. Attend all Release and Iteration planning meetings.
    - This is where the basic functionally of the assigned Story Cards are firmed up and explained. This provides a good overview or vision of the new functionality to be implemented.
    - Attend the daily Stand Up meeting to keep your finger on the pulse of the Iteration.
2. Create a documentation Task for each Story card in the Iteration.
    - By creating and thereby owning a Task, we receive notifications of any changes to related artifacts such as Story Cards, Use Cases, and Other Requirements etc.
3. Review the Specifications (Features, Requirements, Story Cards, etc.)
    - The specifications define the system and can be written as Features, Use Cases, Other Requirements or Story Cards. Each specification contains vital information to be included in the documentation and must be reviewed.
    - The specifications that you review will depend upon your development process. At Rally, we define Story Cards and associate Requirements to the Story Cards to further elaborate them. In our case, we would need to review both the Story Cards and the Requirements.
    - When the documentation team reviews the specifications, we begin by accessing the Iteration Status view to review the Story Cards. From here we can click the hyperlinked Requirements directly from a Story Card. Once we have the requirement display in front of us, we simply scroll to the bottom of the page to view all related dependents and dependencies for the requirement.
4. Track the progress of the Iteration Story Cards
    - We use the Iteration Status View to track the "Status" of each Story Card. When the Story Card Status displays "completed", this indicates that the feature

or functionality of the Story Card is relatively stable, development has completed all related Tasks and documentation can begin.

5. Verify Software Builds
    - When a Story Card in marked as "completed", documentation team members log onto the latest test build for the Iteration and verify the functionality of the software represented in the build against the requirements. Any discrepancies are addressed with the Product Manager, and requirements are updated as a result of these discussions.
6. Stay apprised of requirement and story card changes
    - Use the My Home tab, My Notifications to view changes to artifacts that could affect the documentation tasks. Any relevant notification is researched by viewing the changed artifact.
7. Test Case Attachment
    - We keep a copy of all our Context Sensitive Help ID numbers and topics in Rally as an attachment to a single Test Case. This Test Case allows us to verify all context sensitive help using one Test Case, and the attachment makes it easy for the tester to determine if the correct help topic is displayed. The documentation team updates the attachment with any updates for the Iteration.
8. Assess Defect Status
    - Rally presents a saved Quick View on the Defects tab that will query for any defects that affect documentation or the release notes. During development and testing, any defects that might impact documentation have been checked by team members as such. (There are two separate check boxes on a Defect entitled Affects Doc, and Release Notes)
    - As we get closer to the end of the Release, this view becomes very important for quickly identifying those issues that may need specific documentation. We investigate each Defect returned by this Quick View, update the documentation as necessary and add a note to the defect regarding any documentation action taken. Product Management then reviews the notes and clears the check mark on the Defect.
9. Create Release Notes
    - We use the same Defect Quick view to identify those issues that should appear in our release notes. Our release notes include a discussion on the new functionality, any known issues and issues that have been resolved.
10. Release the new functionality, and do it all over again! Iterate, Iterate, Iterate.