generic logo
company

Getting Stories from Development to QC in an Organized, Effective, and Clear Manner (Pre-Release)

Author	**Date**	**Comment**
Nathan Rasch	ABC 20XX	Document inception

Table of Contents

Getting Stories from Development to QC (Pre-Release)

Background/Executive Summary

The current method of moving completed stories from feature branches to develop to QC is problematic. Over the past few sprints we've had myriad issues such as application versioning errors, conflicting stories from Product that were at odds with each other, and an inability for the QC team to quickly and accurately ascertain what was contained in a build.

These types of issues have led to a great deal of frustration, confusion, and many hours of time spent trying to mitigate them.

Going forward we need a process that is simple, addresses the issues outlined above, and is scalable as our team grows.

Scope

- For now the assumed scope of this document is for the **ProjectX** product and code base

Goals

- Set Up a repeatable and clearly defined release process
- Build the process to the lowest common denominator as far as team member knowledge, skill, and/or motivation is concerned
- Give any team member a clear view port into what is in any build and/or environment
- Control when items are merged into the **'develop'** branch
- Ensure what does move into the **'develop'** branch is correct
- Give the QC a static target to aim at for testing purposes, and make them part of the process for moving that target

Assumptions

- We want and need a 'single source of truth' that anyone can check to see which stories are in the current branch/build/environment
    - For the **ProjectX** code base this is the '**CHANGELOG.md**' file
- We want to control the cadence at which items are merged into the **develop** branch in order to assure quality, adherence to process, balance QC workload, and enable a macro view of work being performed
- We want to control who can merge items into the **develop** branch.

Process

1. Developer (dev) pulls story from the backlog
2. Dev creates feature branch for story from **'develop'** branch

    a. (Frankly we don’t care what all the dev does in this area of the process. They can make 100 feature branches if that makes sense for them. We don’t want to impinge in any way on whatever creative process they need to solve problems and write great code.)

3. Dev writes code and/or whatever tasks are needed for the story including updating the **CHANGELOG.md** file which is mandatory
4. Once the story is complete the dev submits their changes for peer review in the form of a Pull Request (PR)
5. Once a story passes the PR it is now a candidate for merge approval (MA) into the **develop** branch
6. The MA group—which consists of members from both dev and QC—approve the story for merging into the **develop** branch
7. Once approved the story is merged into the **develop** branch
8. Jenkins creates an updated build of the develop branch, deploys the build to the dev-www.ProjectX.com environment area, and adds a notification to the **#deploy-notify-ia** Slack channel
9. The QC team can now—if they wish—pull the **v-number** from Jenkins and deploy the develop branch build to whatever environment they wish for further testing
10. The QC team—or anyone else for that matter—can browse to a special URL for any non-production environment that prints out the contents of the **CHANGELOG.md** file, so that exactly what’s included in the build/environment is clear and transparent

CHANGELOG.md Modifications

- At the top of the **CHANGELOG.md** file add the following section:

    ## UNRELEASED
    ### Added

    ### Changed

    ### Fixed

    ### Removed

- When a dev makes changes to the **CHANGELOG.md** file as part of the story, they add their story details into the ‘*UNRELEASED*’ section of the file’s contents.

- At the end of the Sprint the Release Team will move the details for each story that is part of that release out of the ‘*UNRELEASED*’ section and into the appropriate release notes area.

- Example:

    ## UNRELEASED
    ### Added

```
- IA2-111 Example finished story one, but not approved for any particular release…

### Changed
- IA2-222 Example finished story two, but not approved for any particular release…


## [2.2.12] - 2016-12-15
### Added
- IA2-111 Example approved story one...

### Change
- IA2-222 Example approved story two...
```

Initial Merge Approval Team Recommendations

- Person One
- Person Two
- Person Three

Initial Develop Branch Merge Permission Recipient Recommendations

- Person One
- Person Two
- Person Three