Processes and Checks

Git Guidelines:
https://docs.google.com/presentation/d/1qvXHfhmnmP2Nm4MDqO9hWnKv8SYtCgQTpaL9b6IyrH4/edit#slide=id.p4

Category	Process	Check
Versioning	• Every release should have a tag (x.xx) • Every push to production should have a NEW tag. ○ If a hotfix needed on server then go back to local, replicate the fix and then push with new version number • Production checkouts should only happen via Git tags instead of master branch ○ git checkout x.xx • Only project managers should be tagging a version (x.xx) ○ Major overhauls increase the major number • Tag should contain description on what was included in a release.	fabric that checks prod directories and whether a git repo is checked out from tag.
Deployment	All production deployments should happen via Fabric (we can have a “template django project” that contains fabric file)	bash history on prod server for “git checkout”
Branching	Should be a ‘develop’ branch	Check project branches
	All Pull Requests go into ‘develop’ branch	Check every PRs that target is develop branch
	Once a branch is merged into ‘develop’ the original branch should be deleted	Check “diffs” between branch and develop is 0. Check initial commit after branch from develop does not exist in develop.
	When develop is merged to master a new version is created	

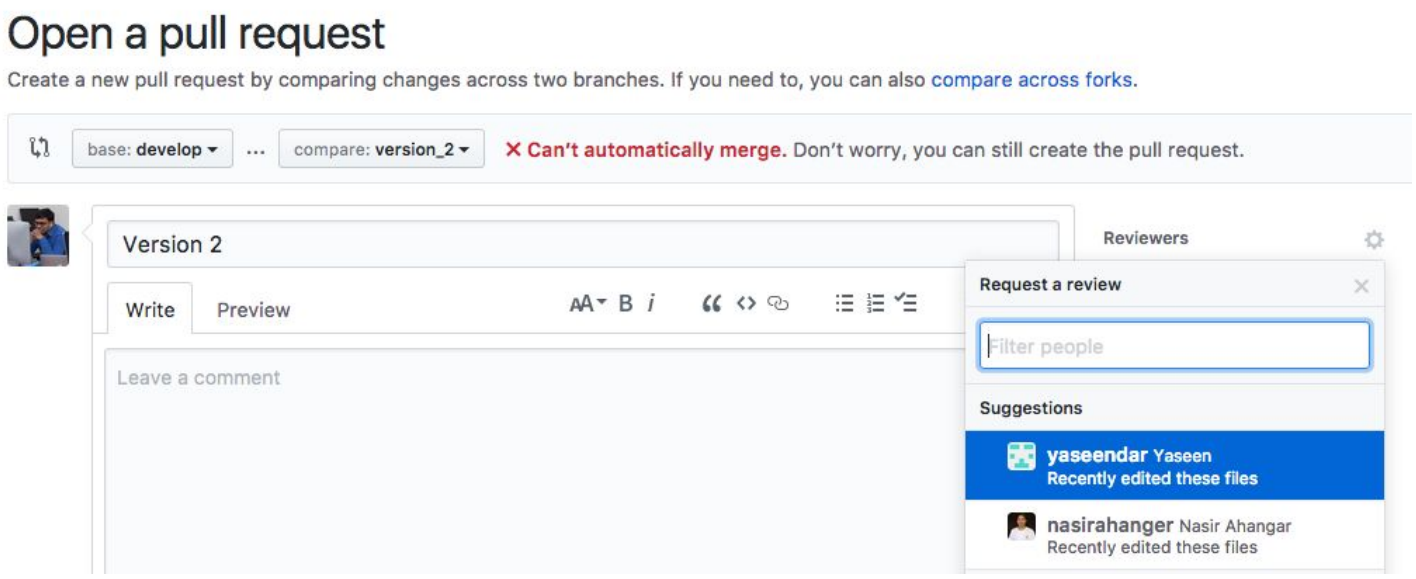
	Branches should not live more than 3 weeks.	Check for each branch start-now (author - notify)
	Only PM can push to develop. nobody else can	
	Should not contain name of the person but contain the JIRA ID or some description of feature or bug. Feature Branch naming: **feature-**<JIRA-ID>-<2-3 word Description> or **feature-**<2-3 word Description> *WL-13-feature-password-reset* *feature-password-reset* For Bug branches naming convention: **bug-**<JIRA-ID>-<2-3 word Description> *WL-55-bug-email-text* *bug-email-text* For the “Merge” branch (when you get conflict - not green symbol - you merge develop into feature and create a new branch+PR) feature-develop-<JIRA-ID>-<2-3 word Description> or **feature-develop-**<2-3 word Description>	

Commit Message	- if you get a merge conflict - add the files changed in the commit message
	
	`Merge conflict on <target> branch into <my branch>`
	`  - <fixed file path 1 with conflicts>`
	`  - <fixed file path 2 with`	

	conflicts> - ….	
Code Review	Code reviews are done through GitHub PR - only for new features - If there is a client deadline/date for feature to be “demoed” - the PR should be sent at least 2 days before.	
Code Review (new Features)	Dev creates a pull requests. The title of the PR should be same as the branch name as described above and specifies following exactly as below per line, in the initial comment: - @tag1 @tag2 (at least 2 devs to review.) - Put a JIRA case ID or link to some doc explaining the feature/requirement - Link to dev server where the reviewer and go and look the working feature - Test cases passing link (Jenkins - GitHub)	Automated script to check this
Code Review (new Features)	This is done when the first initial pull request is sent. Both Reviewer provide final scores as a comment (in addition to feedback provided inline). Block atleast 30mins or more when doing the review (senior devs aim to spend atleast 5-6 hours every week on code reviews): - **Meets requirements**: 0..5 [if 0=then no requirements met, 3=some requirements met, 5=all requirements met .....doesn’t do validation of form….] - **Code Quality**: *-5,-4..0,..4,5* [....-1 - Poor or 0=Good… then specify what could have been done to make it “Best”=5, how simple and elegant. performance… data models, modular..reusing existing code..	Automated script to collate the scores per dev and reviewer

	maintainable.. extensible (use of settings etc)] - **Test Coverage**: -1, 0, 1, 2 [-1 if no tests, 0 if < 50% coverage, 1 if <75 % coverage, 2 = 100% test coverage] - ~~**Applied Code Conventions**: -1 for every violation or 1 if all good [...api.py, views.py, function lines < 100, presence of doc string on every function/class...consistency, Errors/Exceptions/Logging.. no large blocks of commented out code with out reason]~~ - ~~**Python Code conventions**: -1 for every violation or 1 if all good [...idiomatic python, PEP..]~~ - **Bonus**: +1 for every good deed (e.g. above and beyond the expectations of the requirements. adding form validation even if not part of specs or you used some interesting approach to solve.. e.g use decorator some new pattern) [... explain...why..]	
Code Review (new Features)	In next comment, reviewer will add a :thumbsup: 👍 if code is ready to get merged or :thumbsdown: 👎 if it needs more changes.	Automated script check
Code Review (new Features)	Timeline - maximum 2 business days or early if we have immediate deadline - one person review and fine to deploy for testing (by internal or external product owner)	Automated script check
Code Review (new Features)	If the code needs more changes.. then after dev receives a :thumbsdown: then the changes are commited to that branch and dev @tag1 and @tag2 both reviews to re-review the code. Next review you just gives :thumbsup: or :thumbsdown: no scoring.	
Code Review	https://docs.google.com/presentation/d/1qvXHfhmnmP2Nm4MDqO9hWnKv8SYtCgQTpaL9b6IyrH4/edit#slide=id.gb6d71453d_0_0	

Code Review (Bugs/Quick fixes)	For Bugs, quick fixes, we will have a “fast track” code review process where you will still create a Pull Request and assign only one other developer in your timezone. If you tag the person before 4pm local time, the review needs to be completed by EOD, else next day.	
Code Review (Bugs/Quick fixes)	**Syntax of Pull Request** - @tag1 @tag2 (at least 2 devs to review.) - Put a JIRA case ID of the bug In next comment, reviewer will add a :thumbsup: 👍 if code is ready to get merged or :thumbsdown: 👎 if it needs more changes.	
Applied Code Conventions	1. Doc Strings 2. Error Handling 3. Logging	
Python Code Conventions	We Need To Follow PEP - 8	


Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: develop ... compare: version_2 ✕ Can't automatically merge. Don't worry, you can still create the pull request.

Version 2

Leave a comment

Reviewers

Request a review

Filter people

Suggestions

yaseendar Yaseen
Recently edited these files

nasirahanger Nasir Ahangar
Recently edited these files