Centralized Workflow				
	master	default development branch		
		created on cloning repo.	_	
	origin	Origin refers to the parent repo.	Ma	ster
git clone ssh://user@host/path/to/repo.git				
git status				1
git add/rm <files></files>				
git commit -m "****"				
git push origin master				
If Updates were rejected>	git pullrebase origin master		\sim	
	git status		Both pushed to central reposito	
	If Unmerged paths>	git add/rm <files></files>	central repositor	у
		git rebase continue		
		If you have no clue what to do>	git rebaseabort	
		git push origin master		

Feature Branch Workflow			
	master	master never contains broken code!!	
	<feature1-branch></feature1-branch>	Cada assisso DEFODE audia-	
	<feature2-branch></feature2-branch>	Code review BEFORE pulling	
	<featuren-branch></featuren-branch>	features into master	
git clone ssh://user@host/path/to/repo.git			
git checkout -b <feature1-branch> git checkout <feature1-branch></feature1-branch></feature1-branch>	Request creating a new branch with -b or if branch is already created, then do not use -b		—
git branch -v	Show which branch you are working on.		
git status			
git add/rm <files></files>			
git commit			
git push -u origin <feature1-branch></feature1-branch>			
When done with feature, send pull request to team>	git push		
	Review code by 3rd party>	git checkout <feature1-branch></feature1-branch>	
		Make changes as needed, then publish>	git checkout master
		1) ensure master up to date.	git pull
		2) pull new feature into master	git pull origin <feature1-branch></feature1-branch>
			git push

	Same as Feature Branch Workflow		
Gitflow Workflow	except roles are assigned to feature		
	branches.		
	master	master contains abridged version with TAGS.	Tag with version number
	hotfix (short lived)	Only branch forked directly from master	Make fix, merge into master, and version
	release (short lived)	only branch formed an early from master	make my menge into master, and version
One time setup	develop	develop contains all history of project.	
git branch develop	ucvelop	action contains an instary of project.	
git push -u origin develop		Master Hotfix Release Develop	eature Feature
Each Developer			
git clone ssh://user@host/path/to/repo.git			
git checkout -b develop origin/develop		v0.1	v1.0
Developer works on feature		↓	+
git checkout -b <feature> develop</feature>	<feature1></feature1>	0	
git status	<feature2></feature2>		
git add/rm <files></files>	<feature2></feature2>		
git commit -m "*****"	riedturenz	_	
gic commit -m	git pull origin develop		
Finish feature ->>	git checkout develop		
	git mergeno-ff <feature></feature>		
Merge <feature> to develop branch</feature>		· · · · · · · · · · · · · · · · · · ·	
Delete <feature> branch AFTER you verify merge</feature>	git push Conflicts ??>	Same as Centralized Workflow	
worked.		Same as Centralized Workhow	
	git branch -d <feature></feature>		
	Ready to release??>	git checkout -b release-<#.#> develop	Feature frozen
		Update documents, test, and prepare for release	
Found a Bug!! Checkout from master.		Release ->	git checkout master
git checkout -b issue-<####> master			git mergeno-ff release-<#.#>
Fix the bug and merge fix to master and tag.			git push
git checkout master			git checkout develop
git merge no-ff issue-<####>			git mergeno-ff release-<#.#>
git push			git push
git tag -a <#.#.###> -m "****** master			git branch -d release-<#.#>
git pushtags			git tag -a <#.#> -m "****** master
Update develop branch if no release branch exists			git pushtags
git checkout develop			Done
git mergeno-ff issue-<###>			
git push			
git branch -d issue-<####>			
Jpdate release branch if release branch exists			
git checkout release-<###>			
git mergeno-ff issue-<####>			
git merge no-ff issue-<####> git push			