

1 WORKFLOW FIRST CHARACTERISATION

By attempting to understand the kinds of stipulation included in the Git workflows' descriptions, we can begin to create a common framework in which they can all be described. Once characterised in that common framework, their differences and similarities would become much easier to see. Gaps in the descriptions of individual workflows would be more apparent, and families of workflows that might exist, and be useful, but which at present are troublesome to be identified, would be discovered more easily.

Having gathered the specifications for the workflows identified for the research, the next step is to describe each workflow using a common set of features that can be applied to all the workflows in the list.

Following this process, a number of groups of features was identified, i.e., general, branching, operations and others, that provide the initial structure for our framework. The result is presented in Table 1. In this table, a ✓ indicates a type of Git interaction that must be followed if the workflow is to be adhered to, a ✗ indicates an interaction with the Git repository that is proscribed by the workflow, a combined ✓✗ indicates that any of the available options are acceptable, and a ? symbol means that no information is given about the use of this kind of interaction in the workflow documentation.

Table 1: Workflow characteristics

Workflow	Branching model				Operations			Tags	Other features	Recommended scenarios
	Single (S) / Multiple (M) branches	Change branch	Release branch	Naming convention	Clone / Fork	Merge	Rebase	Cherry pick		
Backcountry	M	Feature	Production	✓	C	✓	?	✓X	✓	Large projects, multiple versions
BOINC	M	Feature	Release	✓	C/F	✓	?	?	?	Continuous deploy
Cactus model	M	Any	Release, Master	X	C/F	X	✓	✓	✓	Not open source nor embedded prog.
Centralised	S	Master	Master	X	C	X	✓	X	?	Small projects
Dictator and Lieutenants	M	Topic	Master	X	?	✓	✓	?	?	Large, highly-hierarchical projects
Feature branch	M	Any	Master	X	C/F	✓	✓	✓X	✓X	Large projects, CI
Fork & merge	M	Feature	Master	X	F	✓	?	?	✓X	Open source
Git Common	M	Feature	Release	✓	C	✓	✓	X	✓	Continuous development, open source
GitFlow	M	Feature, Topic	Master, Release	✓	C/F	✓(no-f)	✓X	✓X	✓	Scheduled projects, single version
GitHub	M	Any but Master	Master	X	C/F	✓	✓X	✓X	✓X	Continuous deploy
GitLab	M	Feature	Release	✓(link issues)	C	✓	?	✓X	✓	Large projects, multiple versions
GitWaterFlow	M	Feature	Release	✓	C	✓	✓	?	?	Forward porting
IGSTK Flow	M	Topic	Master	X	C	✓	?	?	?	Agile-based
OneFlow	M	Feature	Master	✓	C	✓	?	✓X	✓	Extensive use of tags
PLEFlow	M	Develop	Baseline	✓	C	✓	?	?	✓	Parallel development
Release	M	Feature	Release	X	C	X	X	✓	X	Large projects, multiple versions
Simple Git	M	Feature	Master	X	C	✓(no-f)	✓	X	✓X	Clean merges
Skullcandy's	M	Feature	Master	✓	C	✓	?	✓X	✓	Scrum
Three-Flow	M	Candidate	Release	✓	C	✓(no-f)	✓	✓X	✓	Feature toggles
Trunk-based development	M	Any	Release, Master	X	C	✓	✓X	✓	✓X	No pull requests
WunderFlow	M	Feature	Production	✓	?	✓	✓	?	✓	Testing on multiple parallel versions, showing