

# CI/CD pipelines for UiPath RPA projects

Git Workflows and RPA  
Exploration of Traditional Software Development

Git - Branching Workflows

https://git-scm.com/book/en/v2/Git-Branching-Branching-Workflows

Search entire site...

# git --local-branching-on-the-cheap

About

Documentation

Reference

Book

Videos

External Links

Downloads

Community

This book is available in English.

Full translation available in

azərbaycan dili,

български език,

Deutsch,

Español,

Français,

Ελληνικά,

日本語,

한국어,

Nederlands,

Русский,

Slovenščina,

Tagalog,

Chapters ▾ 2nd Edition

## 3.4 Git Branching - Branching Workflows

### Branching Workflows

Long-Running Branches

Figure 26. A linear view of progressive-stability branching  
It's generally easier to think about them as work silos, where sets of commits graduate to a more stable silo when they're fully tested.

```
graph TD; master --> C1; master --> C2; master --> C3; master --> C4; master --> C5; develop --> C5; develop --> C6; develop --> C7; topic --> C7;
```

```
graph LR; master --> C1; master --> C2; master --> C3; master --> C4; master --> C5; develop --> C5; develop --> C6; develop --> C7; C1 --> C2; C2 --> C3; C3 --> C4; C4 --> C5; C2 --> C3; C3 --> C4; C4 --> C5;
```

From the Git Book

Gitflow Workflow | Atlassian Git

https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

Software Development DevOps Agile Microservices

- > Learn Git
- > Beginner
- > Getting started
- Collaborating workflows
  - Syncing (git remote)
  - git fetch
  - git push
  - git pull
- Making a Pull Request
- Using Branches (Git branch)
  - git checkout
  - git merge
  - Merge conflicts
  - Merge strategies
- Comparing Workflows
- Feature Branch Workflow
- Gitflow Workflow

## Feature branches

### Step 1. Create the repository

Each new feature should reside in its own branch, which can be [pushed to the central repository](#) for backup/collaboration. But, instead of branching off of `main`, feature branches use `develop` as their parent branch. When a feature is complete, it gets [merged back into develop](#). Features should never interact directly with `main`.

```
graph LR; Main[Main] --- Develop[Develop]; Develop --- v01[v0.1]; v01 --> v02[v0.2]; v02 --> v10[v1.0]; v01 --- F1(( )); v02 --- F2(( )); v10 --- F3(( )); F1 --- v02; F2 --- v10; F3 --- v10;
```

It's trivial to react on these events

HOME DESIGN DEBUG

Cut Undo  
Save Redo  
Export As  
New File  
Manage Packages  
Manage Entities  
Test Manager  
App/Web Recorder  
Computer Vision  
User Events  
Table Extraction  
UI Explorer  
Remove Unused  
Analyze File  
Export to Excel  
Publish

Project   
FrozenChlorine  
Dependencies (Windows)  
Entities  
Templates  
Objects  
Test Data  
Data  
Documentation  
Framework  
Process  
RnD  
Scripts  
Tests  
.gitattributes  
.gitignore  
AUTHORS.md  
LICENCE.md  
PathKeeper.xaml  
project.json  
StandardCalculator.xaml

Test Explorer   
ParseTermToList  
TestCase\_Unit\_ParseTermToList (4)  
PathKeeper  
TestCase\_PathKeeper  
StandardCalculator  
TestCase\_EndToEnd\_StandardCalculator...  
AdditionOf2Terms  
ClickListOfCharacters  
CloseAllApplications  
InitAllApplications  
InitAllSettings  
Initialization  
KillAllProcesses  
ResetCalculator  
RnD\_gitkeep  
RnD\_StringContainsNegate  
Teardown

Test Results Activity Cov... Descriptor C...  
Test Summary  
Total: 10  
Passed: 0  
Failed: 0



# CI/CE Pipelines make "Publish" obsolete

# Don't gamble towards releases

