camelCase

Development Process

Continuous integration, delivery, and deployment - Git VCS

Introduction to Development Process

- Overview
- Challenges
- Action Plan
  - Continuous Integration, Delivery, Deployment
  - Git as Version Control System
  - Automated Testing
  - Deployment Automation Tool
  - Database migrations

Main Challenges

- Big size application that is complex to deploy and having lot of parts affecting the application

Main Challenges

- Different environments where the team need to deploy the application, which may include many team groups collaborating each other to make successful deployments

Main Challenges

- Manual configuration management of production environments

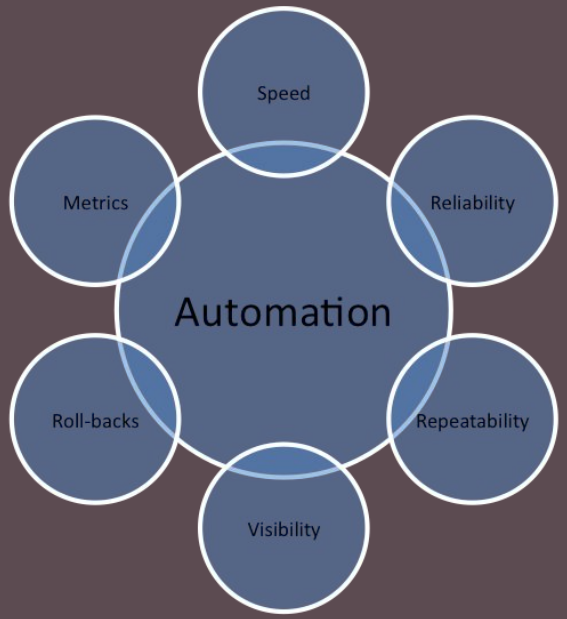
Software Delivery Process

Good software delivery process properties:

1. Repeatable, reliable process for releasing software
2. Automate everything you can
3. Keep everything on version control system
4. Build Quality In (earlier you catch the defect/error the cheaper to fix it)
5. Done means released
6. Everybody is responsible for delivery process
7. Continuous improvement of delivery process

Automated and Frequent Deployment

- Software release goals
- Automated tests and deployment
- Why feedback is important?

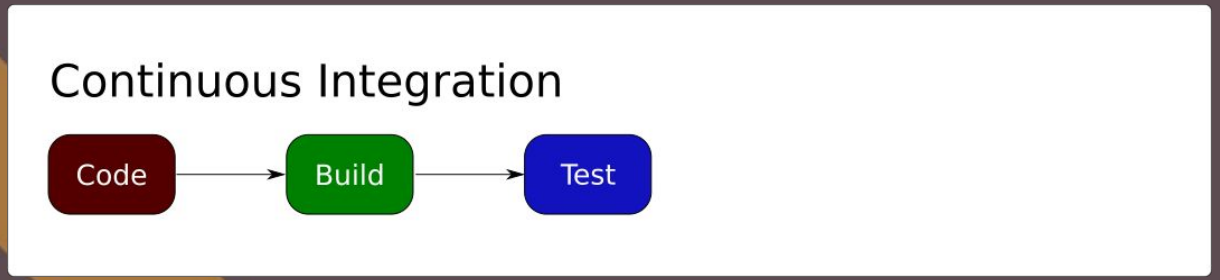


Automated and Frequent Deployment

- What are change types?
- Why early stage tests are important?

Continuous Integration

- What is it?
- What are the benefits?



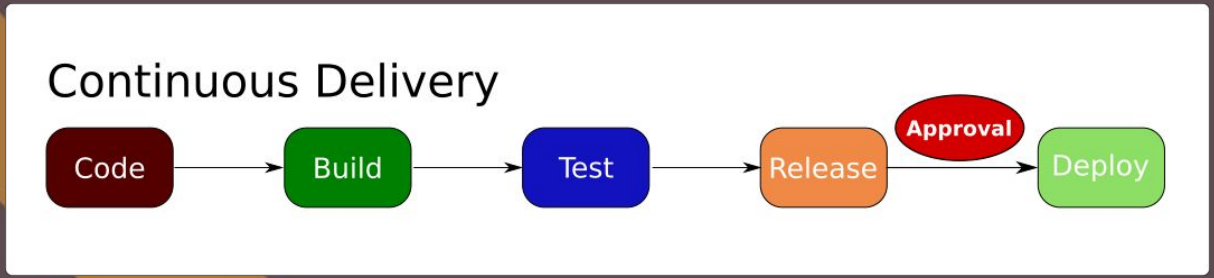
Continuous Integration

What is needed to do CI?

- Version control system
- Automated builds (take care that build scripts have to be treated like code base, tested, refactored, enhanced)
- Team agreement and commitment

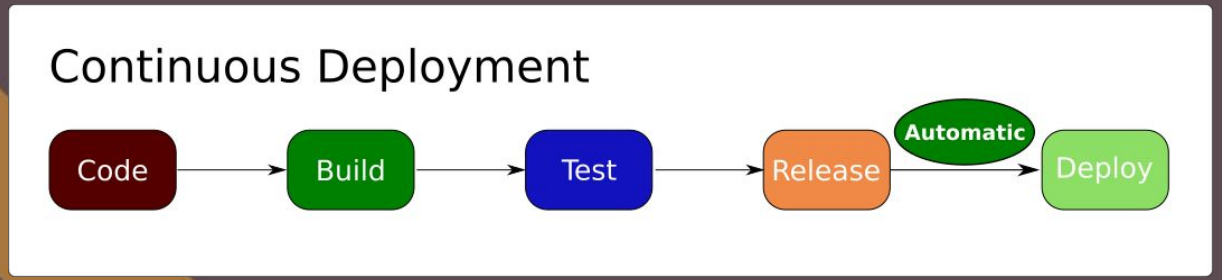
Continuous Delivery

- What is it?
- What are the benefits?



Continuous Deployment

- What is it?
- What are the benefits?

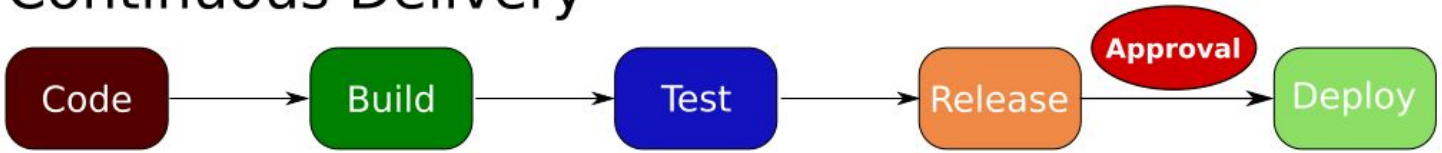


Continuous Integration, Delivery, and Deployment

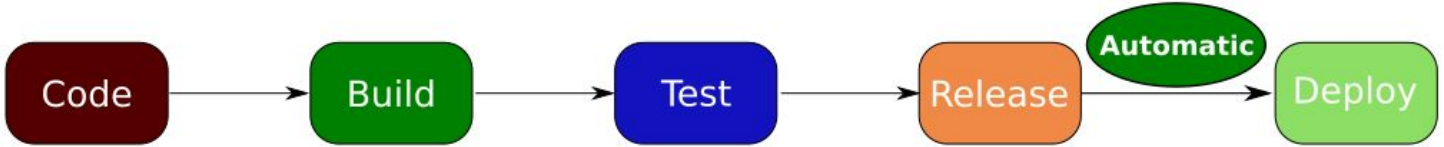
Process	Needs	Benefits
Continuous Integration	- Automated tests for features/enhancements/bug fixes - CI server - Code merge as often as possible	- Less bugs to production due to automated tests - Easy building releases due to integration tests - Testing costs reduced dramatically due to CI server running automated tests - QA team focus on new tests and testing culture instead of repeating same tests again
Continuous Delivery	- Strong CI process - Enough code base coverage - Automated deployment though the manual trigger - Feature flags to turn off incomplete features on production	- Less complexity of software releases - Faster feedback loop with customers
Continuous Deployment	- High quality testing and automated testing suite - Feature flags are essential - Documentation of the deployment process	- Faster development since no need to stop development for making releases - Reduce the risk of releases since they are in small batches - Customer can see improvement in a continuous way, instead of waiting for weeks or months

Continuous Integration

Continuous Delivery



Continuous Deployment



Implementing Continuous Integration

As mentioned before team need 3 main things:

- Version control
- Automated Builds
- Team agreement and commitment

Prerequisites for Continuous Integration

- Merging to main branch regularly
- Create comprehensive test suite
- Keep short build and test process
- Managing your development environment (everything on version control, CM of third-party dependencies, automated tests can run on developers machines)

Essential Practices

- Broken builds
- Run unit testing locally
- Time frame to revert
- Breakage responsibility

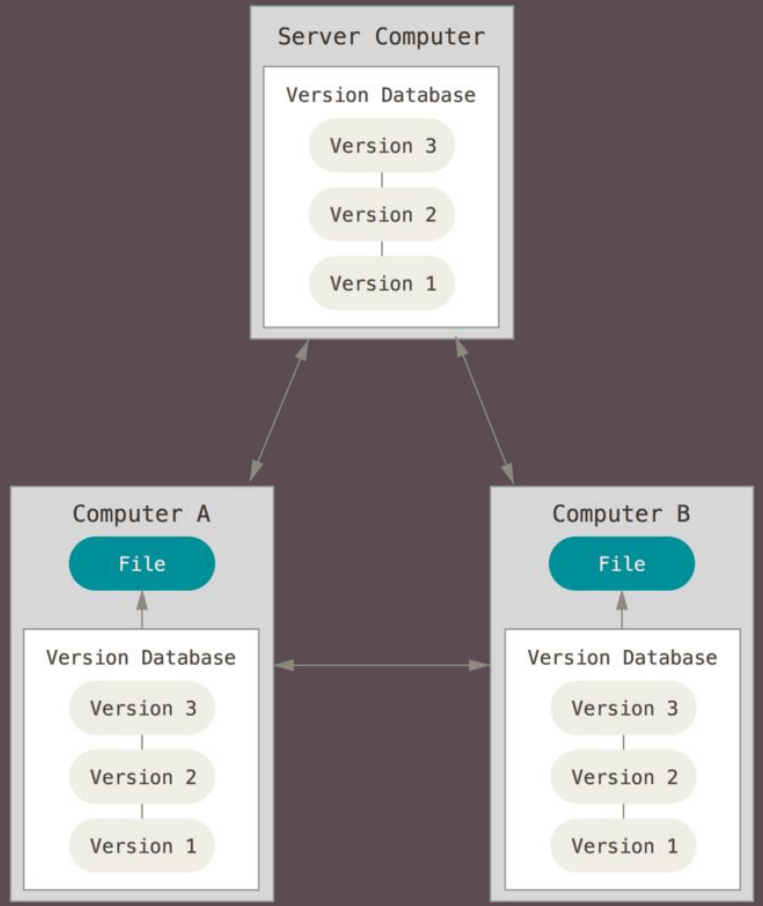
Automated Tests

Tests that are performed by a machine by running test scripts were written earlier by testing person.

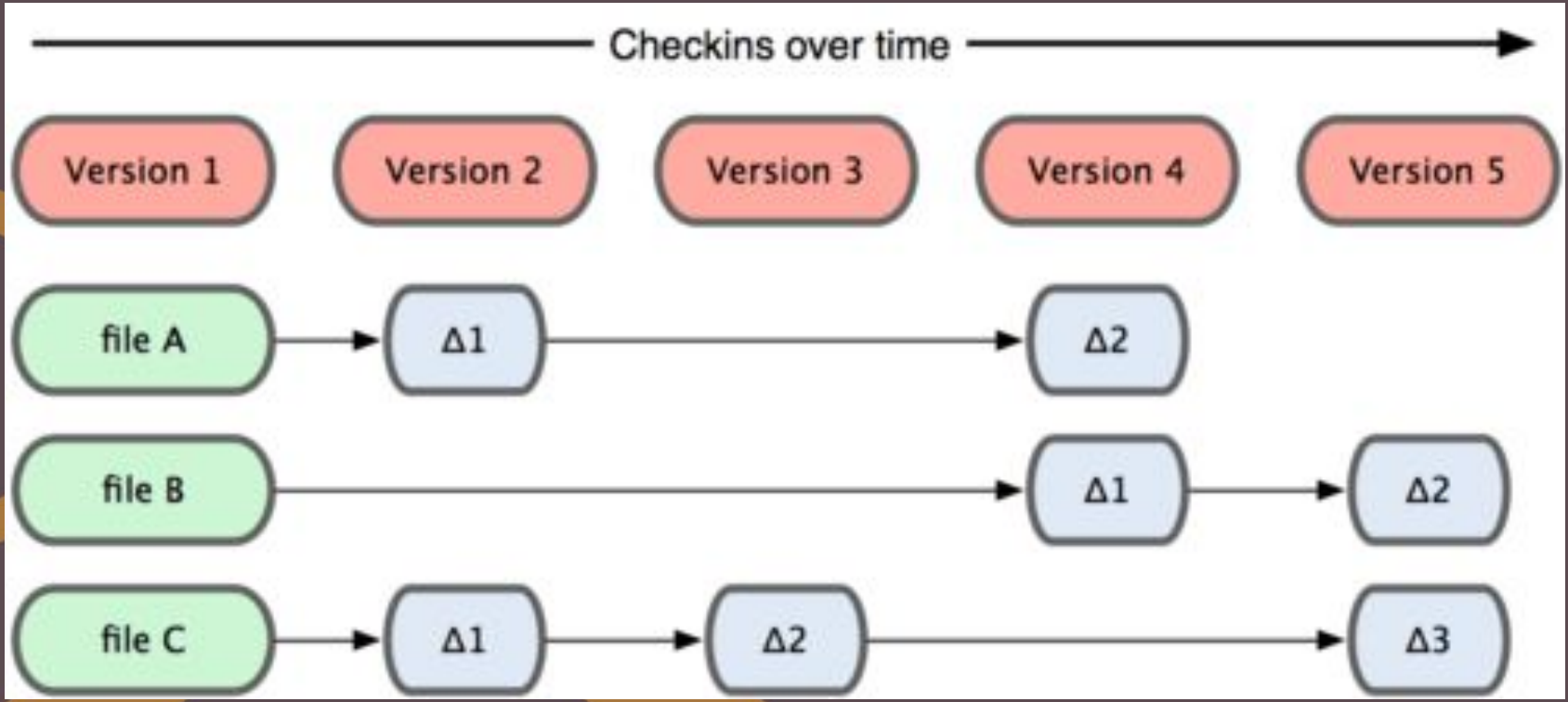
- Unit Tests
- Integration Tests
- Functional Tests
- Acceptance Tests
- Performance Testing
- Smoke Testing

Git Version Control System

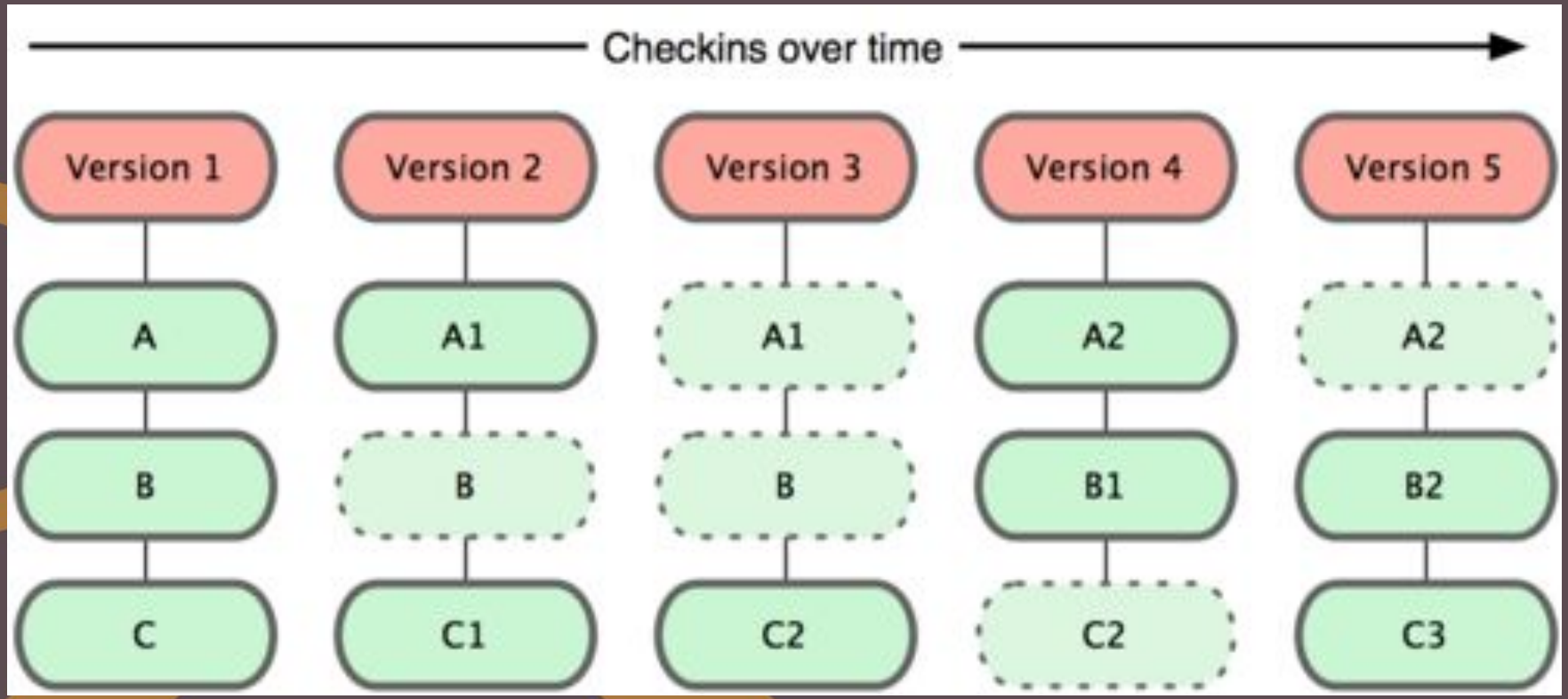
- What is Version Control System?
- What is Git and why is different?



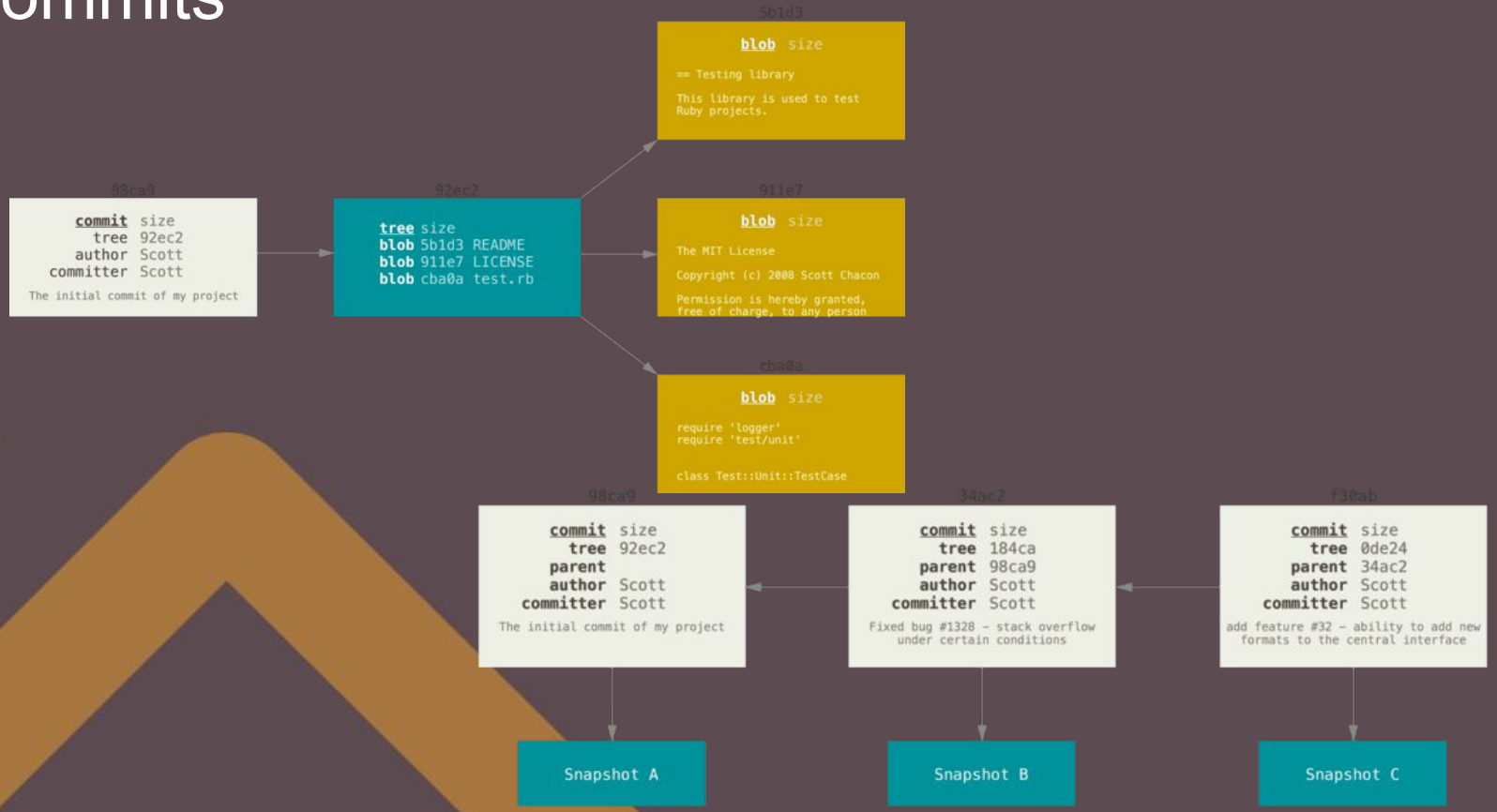
VCS other than Git



Git Snapshot



Git Commits


5b1d3
blob size
== Testing library
This library is used to test Ruby projects.
98ca9
commit size
tree 92ec2
author Scott
committer Scott
The initial commit of my project
92ec2
tree size
blob 5b1d3 README
blob 911e7 LICENSE
blob cba0a test.rb
911e7
blob size
The MIT License
Copyright (c) 2008 Scott Chacon
Permission is hereby granted, free of charge, to any person
cba0a
blob size
require 'logger'
require 'test/unit'
class Test::Unit::TestCase
98ca9
commit size
tree 92ec2
parent
author Scott
committer Scott
The initial commit of my project
34ac2
commit size
tree 184ca
parent 98ca9
author Scott
committer Scott
Fixed bug #1328 - stack overflow under certain conditions
f30ab
commit size
tree 0de24
parent 34ac2
author Scott
committer Scott
add feature #32 - ability to add new formats to the central interface
Snapshot A
Snapshot B
Snapshot C

Git Main Capabilities

- Changes
- Branching
- Conflicts
- History
- Tags (releases)
- Roll back
- Alerts
- Code review
- Files
- Client

Working with Git

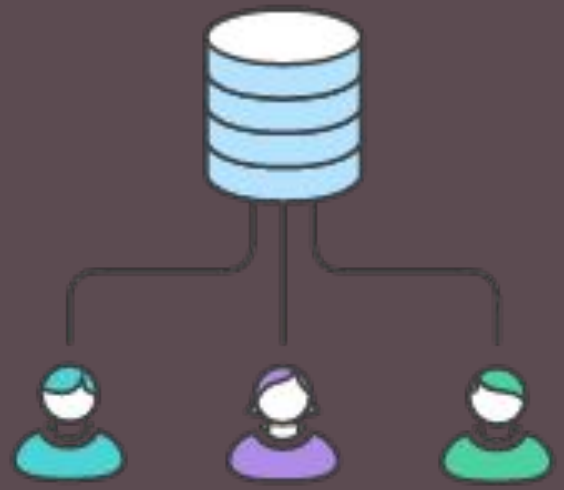
- Initialize local git repository
- Clone from existing repository
- Local branches
- Local processes
- Server processes

Git Workflows

- Centralized workflow
- Feature branching workflow
- Gitflow workflow

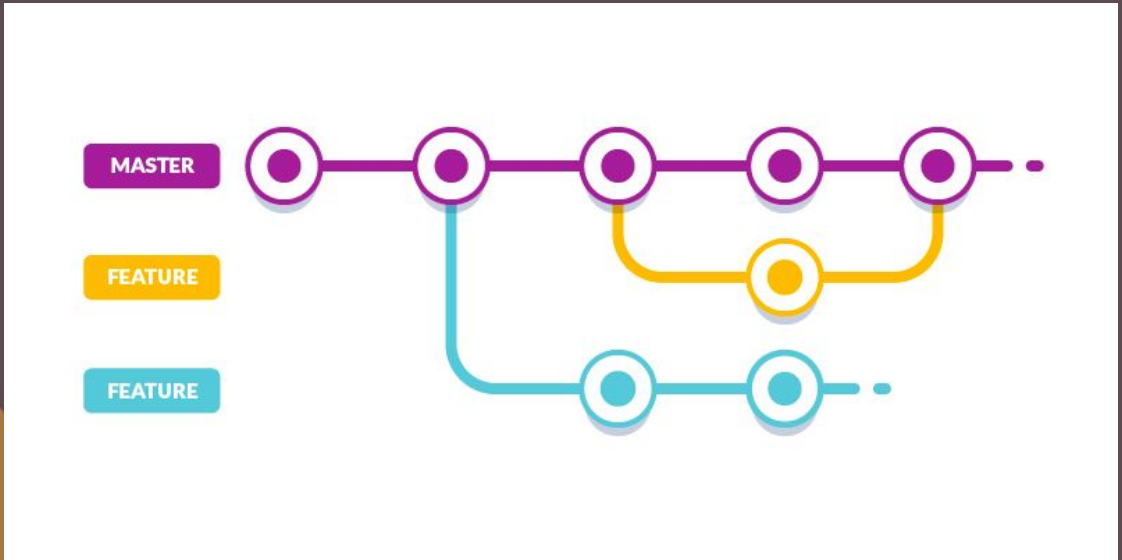
Centralized workflow

- What is the benefit?
- How it works?



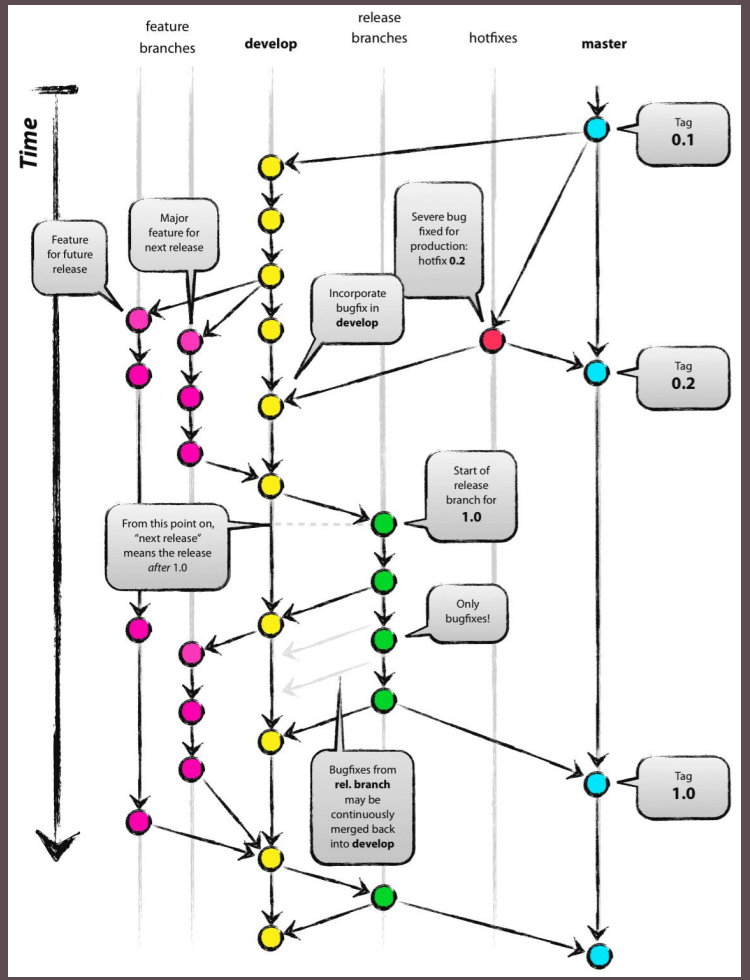
Feature branching workflow

- What is the benefit?
- How it works?



Gitflow workflow

- This is an important workflow for big projects
- To be discussed in details

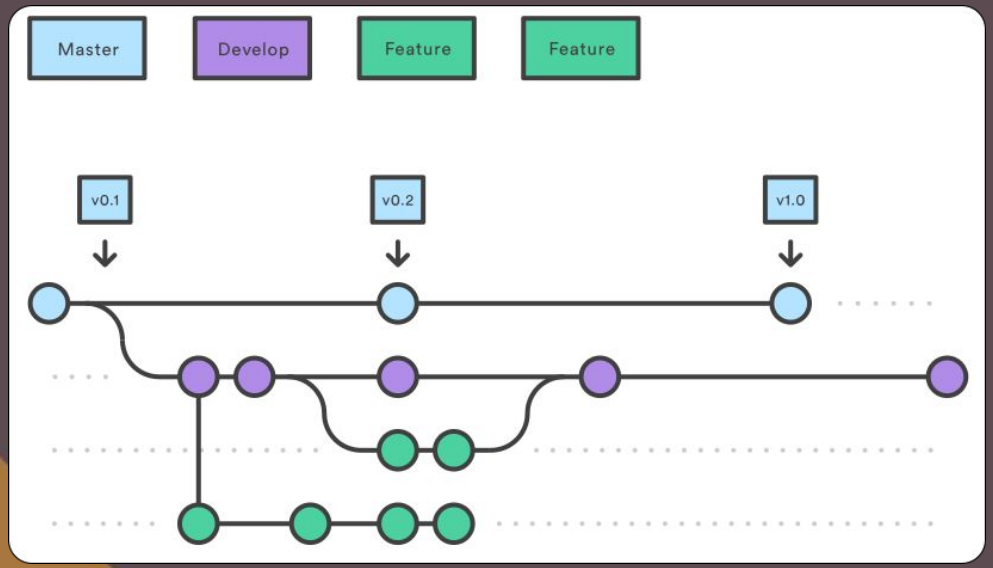


Gitflow

- First introduced by Vincent Driessen
- Depends on 2 main branches develop and master
- Master branch HEAD represents production ready state
- Develop branch HEAD represents latest delivered development changes
- When source code on develop reach stable state it should be merged to master through workflow to be described and tagged with release number
- Workflow process supporting branches are Feature, Release, Hotfix branches
  - Feature branch: no special naming convention
  - Release branch: named like with release-*
  - Hotfix branch: named like hotfix-*

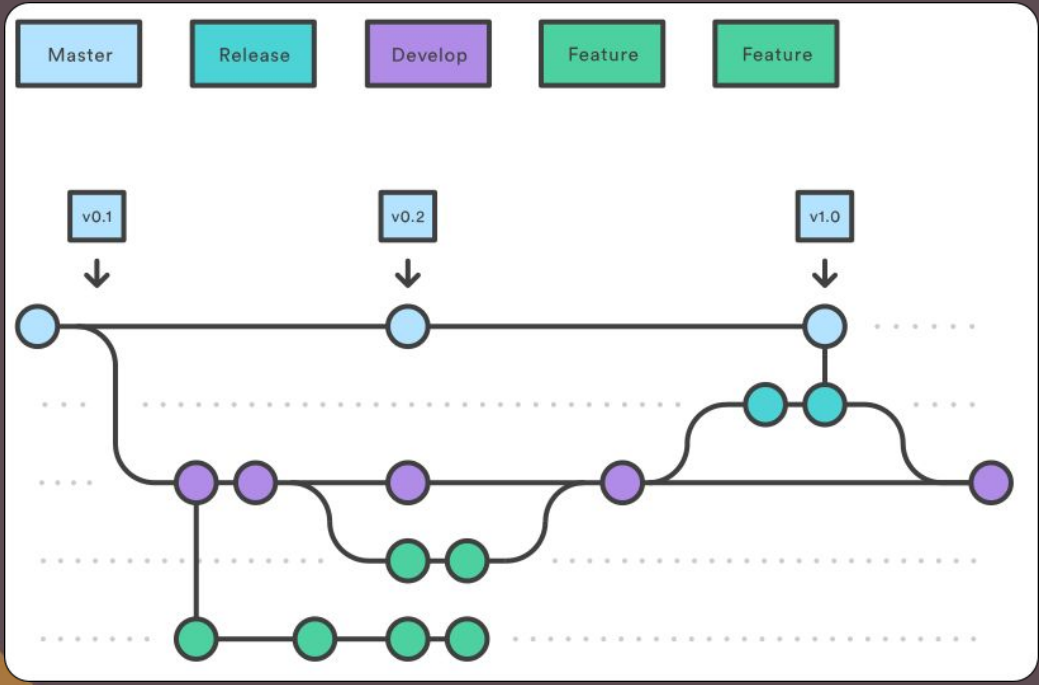
Gitflow - New Feature

- Create feature branch from develop
- Do all changes to feature branch
- Merge feature branch into develop
- Delete feature branch



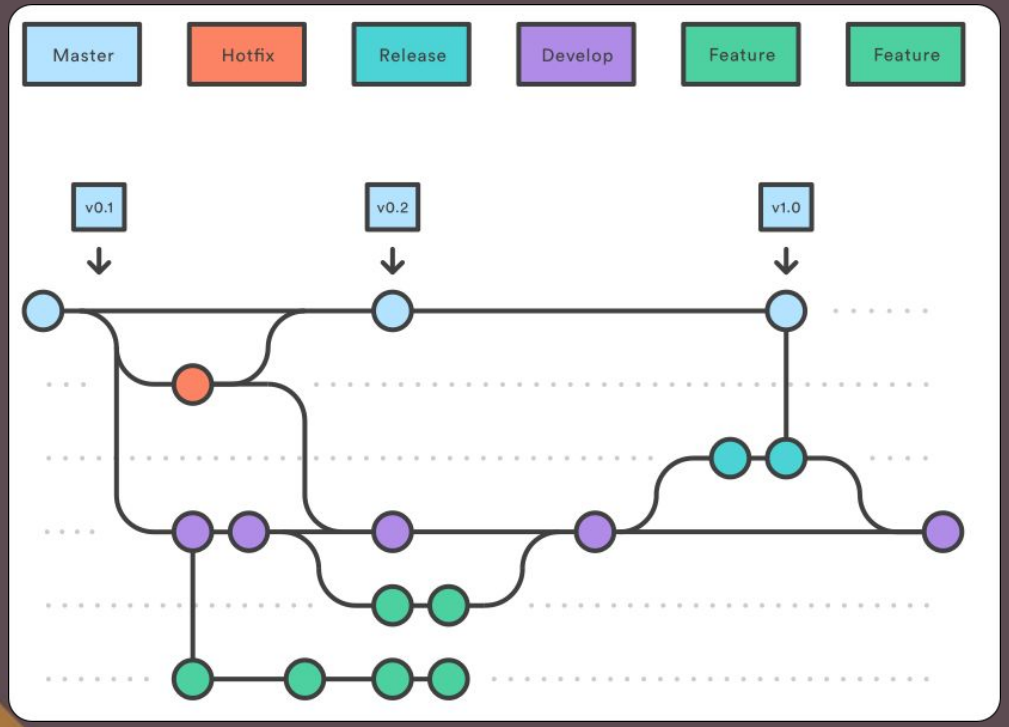
Gitflow - New release

- Create release branch from develop
- No new features added after this point, only bug fixes or documentation generation ...etc.
- Merge release branch back to develop
- Merge release branch to master
- Delete release branch



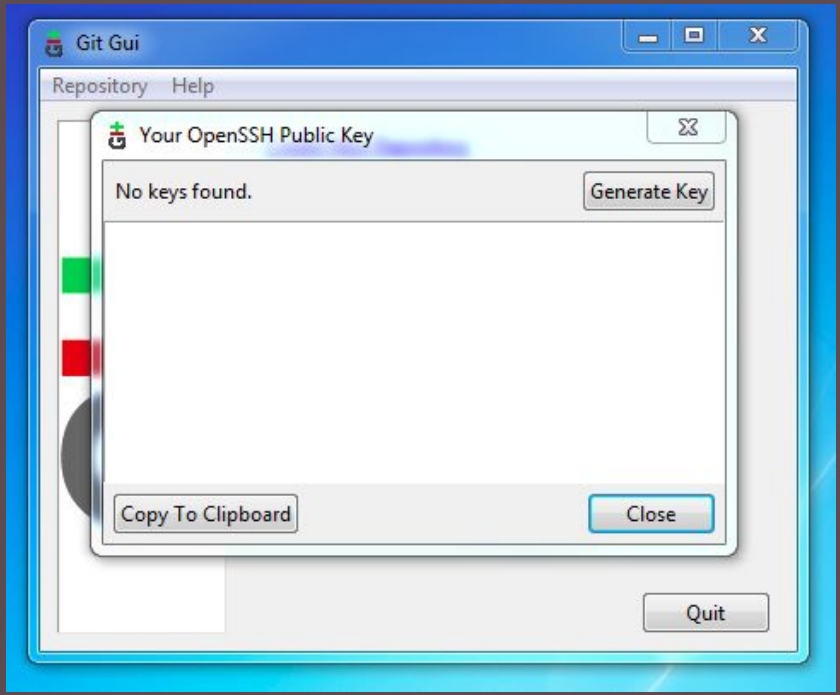
Gitflow - Hotfix

- Create hotfix branch from master
- Do all changes to hotfix branch
- Merge hotfix branch into master
- Merge hotfix branch into develop or current release branch
- Delete hotfix branch



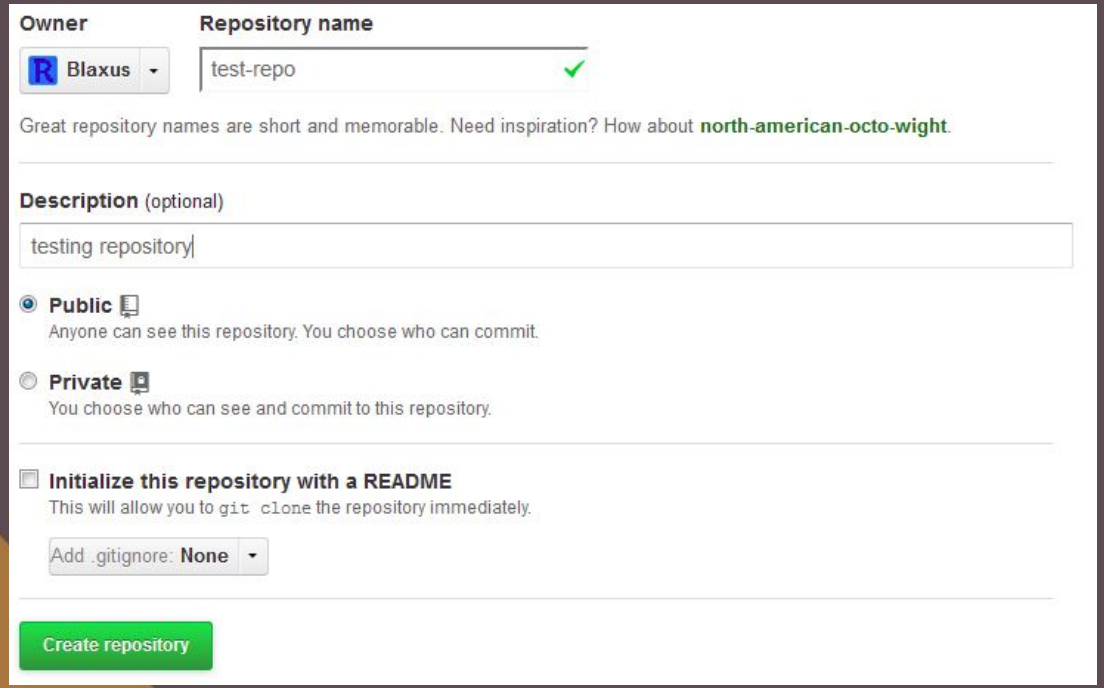
Git Installation

- Download binary from https://gitforwindows.org/
- Generate SSH key from Git GUI
- Add SSH to hosted repository

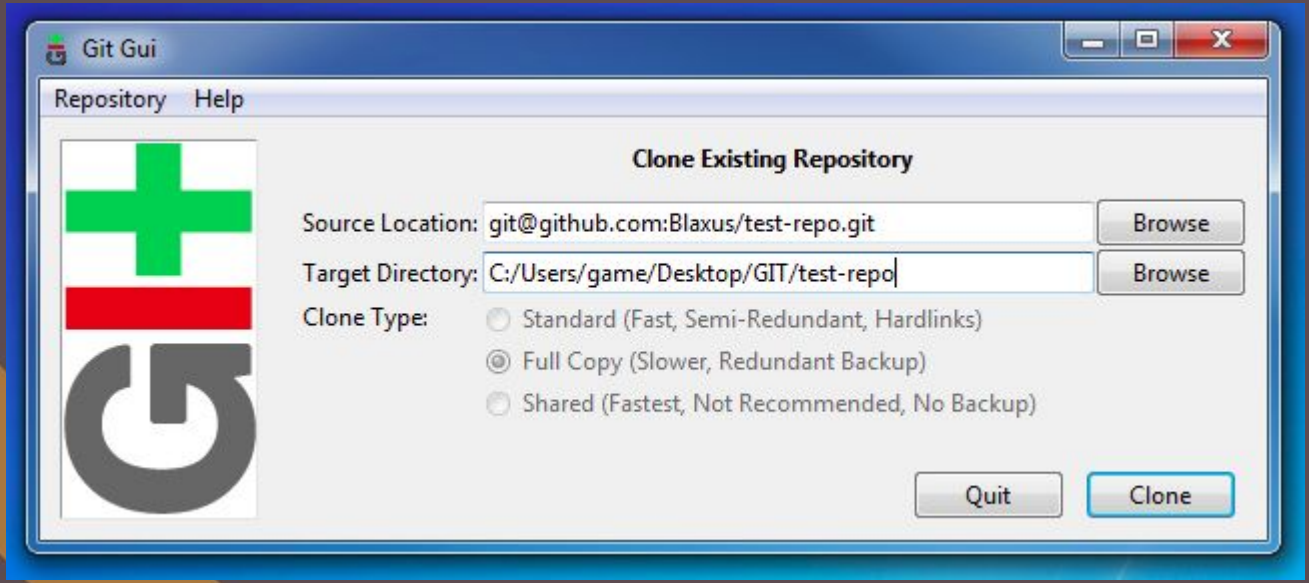


Example Git Repository

- Create repository on github.com



Clone Git Repository



Git Applied Lab

- Please go ahead with small project
- Do the processes discussed earlier in the first session.

camelCase

Thank You