Git Source Control

DevOps Practitioner

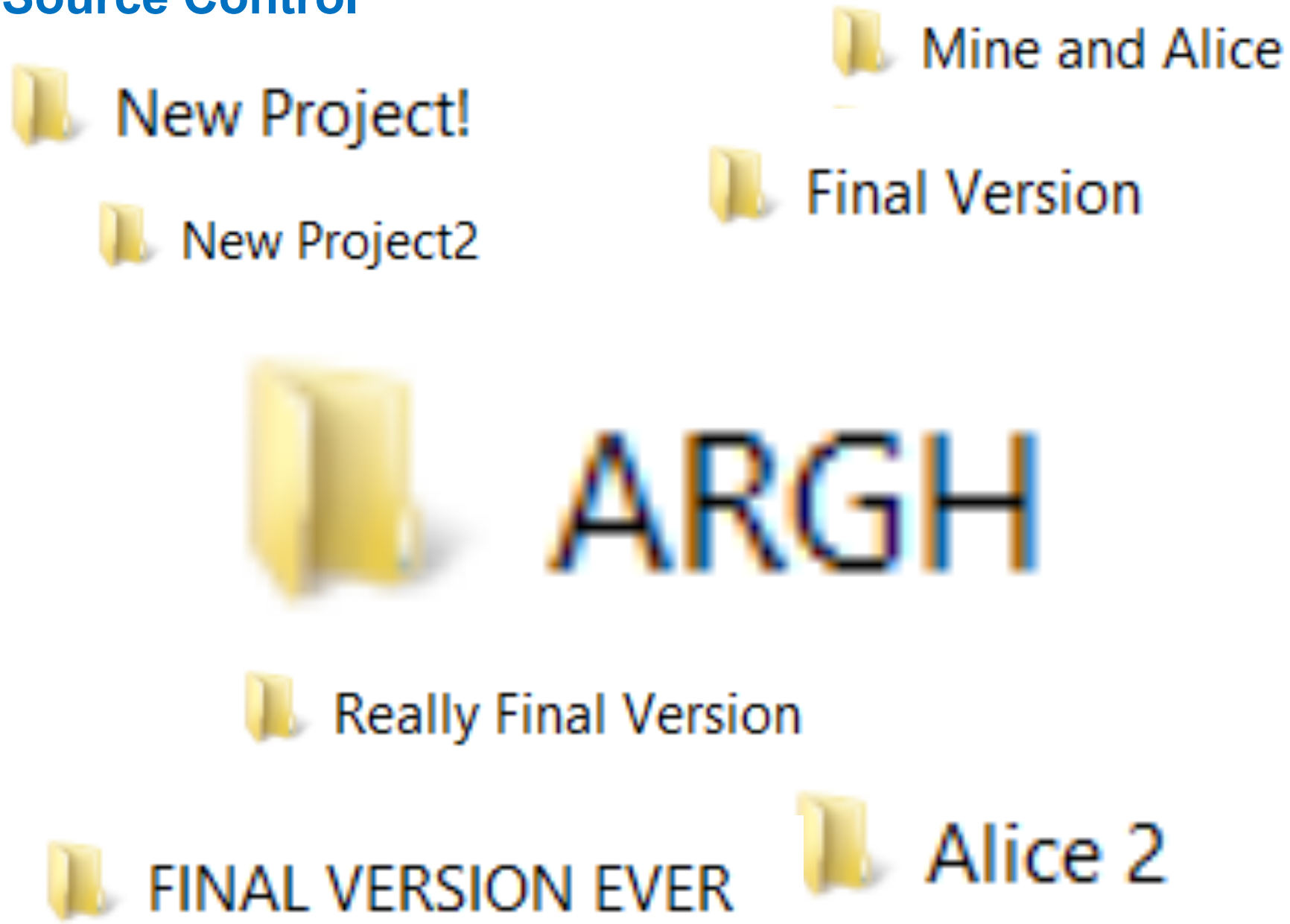
Outline

- **What is Git?**
  - Source control concepts
  - Installing git
- **Git repositories**
  - Cloning repositories
  - Adding files
  - Commit
  - Pushing and pulling files
  - Create and merge a branch
  - The stash and merges
  - Other useful command
- **Alternatives and readings**

Objective

- **By the end of this session you should be able to:**
  - Clone a git repository
  - Add some files
  - Commit changes
  - Create and merge a branch
  - Stash your files
  - View logs

Source Control



Why use source control?

- **Keep track of code and changes**
  - One copy of the code everyone has access to
  - No more mailing around code and confusion trying to integrate it
  - Automated version management

- **Allows for multiple people to edit a single project at the same time**
  - Push changes to the central repository
  - Everyone can pull changes from others
  - Merge together changes in files where there are conflicts

- **Branch code to work on specific parts**
  - Version 2.3 doesn't need to die because someone else wants to look at version 3

Installation of git

- **If you use Ubuntu (or other Debian base GNU/Linux) OS:**

```
$ sudo apt-get install git-core -y

$ git --version
```

- **Other Linux OS:**

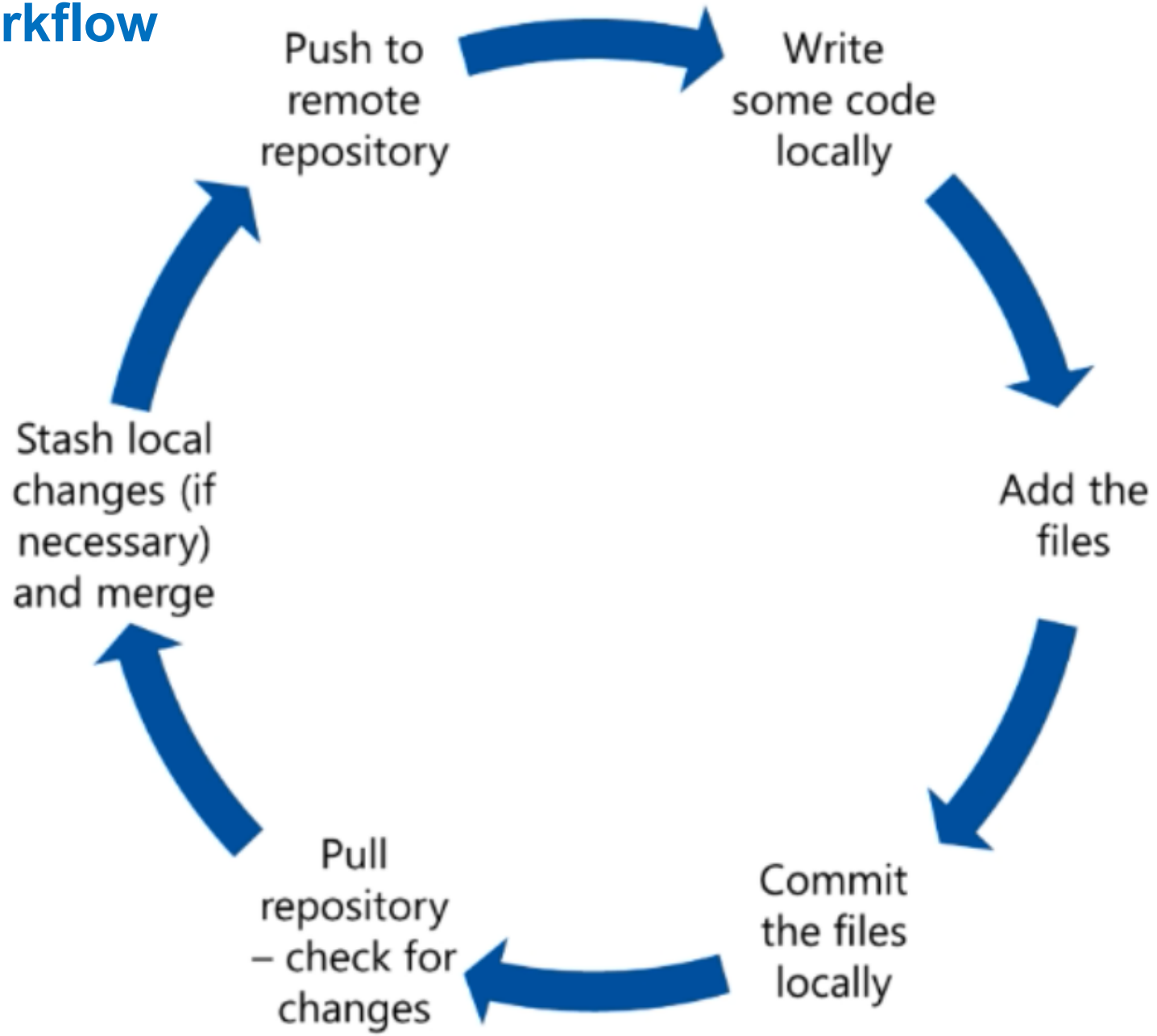
```
$ sudo yum install git-core -y

$ git --version
```

- **Synopsis for git commands:**

```
$ git *
```

Git workflow



Clone your project git repository

- **Git is a popular distributed source control method**
- **Free hosting sites available**
  - Bit Bucket
  - Github

- **To clone your project from git repository:**

```
$ mkdir projectName
$ cd projectName
$ git clone gitUser@git.server.com:yourProject.git
```

Step 1 – 3: Source Control using Git

master

1.

```
$ vim newFileName.txt
$ git add .
$ git commit -m "message"
$ git push origin master
```

2.

```
# change your code file/files
$ git add .
$ git pull
$ git commit -m "message"
$ git push origin master
```

3.

```
# change your code file/files
$ git add .
$ git status
$ git commit -m "message"
$ git push origin master
```

Adding files

- **To add a new file use the ‘add’ argument**

```
# a single file
$ git add specific_file_name.ext

# To add all changed files, deleted and untracked
$ git add .
$ git add --all
```

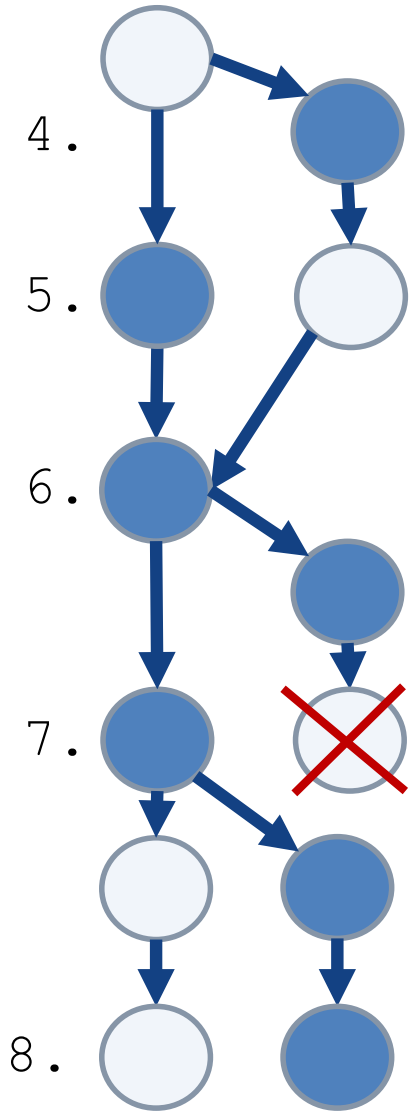
- **Git status will show the newly added file**

```
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

      modified:   DG_02_Git.pptx
```

Step 4 – 8: Branches in Git

Braches allow for teams to add new features and work in parallel without adding 'dangerous' code and changes to the main branch of the repository




```
# create a branch
$ git branch dev
```

```
# switch to a branch
$ git checkout master
```

```
# merge the branch
$ git merge dev
```

```
$ git branch ops
```

```
# delete a branch
$ git branch -d ops
```

```
$ git branch test
```

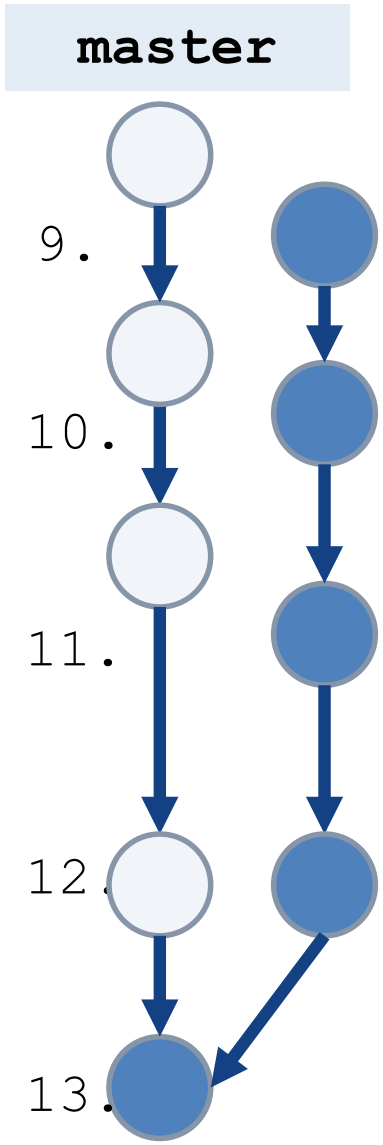
```
# list all branches
$ git branch --list
```

```
# pushes all
# branches to server
$ git push --all
```

```
# Clone to specific
# branch
$ git clone -b BrName
    repositoryAddress
```

```
# push to branch on remote
$ git push --set-upstream origin test
```

Step 9 – 13: Stash and Merge in Git



Stash saves your changes locally, allowing you to pull the latest copy of the repository without overwriting your changes (the changes are stored in a stack)

```
# to stash your changes
$ git stash
```

```
# list all the stashes
$ git stash list
```

```
# apply all stashes to the common
$ git stash apply
```

```
# pull the repository
$ git pull
```

```
# apply changes to working copy
$ git stash pop
```

Other useful commands

```
# To view the history of your changes
$ git log

# To see complete differences at each step
$ git log -p

# To get the overview of the change
$ git log --stat --summary

# To see a tree graph
$ git log --graph --oneline

# To clone specific branch
$ git clone -b branchName repositoryAddress
```

Alternatives

- **Git is not the only source control method out there**
  - Popular due to the open source nature
  - Simple to use
  - Context switching between branches easier
  - Local staging area for commits
  - GUI tools available such as Sourcetree
  - Built in tools in eclipse

- **Subversion (SVN)**
  - Similar idea to git
  - Add new files, commit to the repository
  - Pull files from the repository
  - Tortoise / Rabbit SVN give built in windows context menu options

- **CVS, Mercurial, Bazaar, Fossil, Veracity… many others!**

Exercise

- **Create an account on Bitbucket**
  - Create a git repository
  - Clone it on your machine
  - Add some files
  - Commit them to the repository
  - Create a branch

To read more about git

Books:

- Professional Git (2017. link: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-111928497X.html)
- Jump Start Git (2015. link: https://www.sitepoint.com/premium/books/jump-start-git)
- Git in Practice (2014. link: https://www.manning.com/books/git-in-practice)
- Version Control by Example (2011. link: http://ericsink.com/vcbe/)

Tutorials:

- https://git-scm.com/
- https://www.tutorialspoint.com/git/index.htm
- https://www.atlassian.com/git/tutorials
- https://guides.github.com/activities/hello-world/

Summary

- **What is Git?**
  - Source control concepts
  - Installing git

- **Git repositories**
  - Cloning repositories
  - Adding files
  - Commit
  - Pushing and pulling files
  - Create and merge a branch
  - The stash and merges
  - Other useful command

- **Alternatives and readings**