BEGINNERS PROGRAMMING WORKSHOP

Week 7

3 March 2016

PART 1: VIM RC

VIM RC – CUSTOMIZE YOUR VIM EXPERIENCE

Used to customize Vim between uses

- Commands are entered into Vim rc file
- Vim evokes these commands each time its run

Example Utilities:

- Set tabs to spaces (great for Python!)
- Show line numbers
- Re-route keys
- Turn on syntax highlighting
- Wrap long lines of text (not recommended except for .txt types)

VIM RC – CUSTOMIZE YOUR VIM EXPERIENCE

Example Utilities:

- Set tabs to spaces (great for Python!)
- Show line numbers
- Re-route keys
- Turn on syntax highlighting
- Wrap long lines of text (not recommended except for .txt types)
- Set specific commands for certain filetypes
- Auto-save files given a time constraint

VIM RC – CUSTOMIZE YOUR VIM EXPERIENCE

Location:

- In HOME directory, or ~
- File name: '.vimrc'
  - To open:
    - > ls -la
    - > vim .vimrc

EXAMPLE VIMRC

PART 2: GIT

GIT – MODERN VERSION CONTROL

What is version control?

- Tracks changes to files
- Usually allows for branching
  - Taking one version, splitting it into identical versions, and then allowing for independent changes to either branch
- Often offers remote storage of files

GIT – MODERN VERSION CONTROL

What is GIT?

- Torvalds – the Linux kernel guy – developed Git a few years back
- Git is:
  - Fast
  - Free
  - Fully-featured
  - Easy to use (as far as basics go)
  - Common and growing in popularity

WHY IS IT SO FAST?

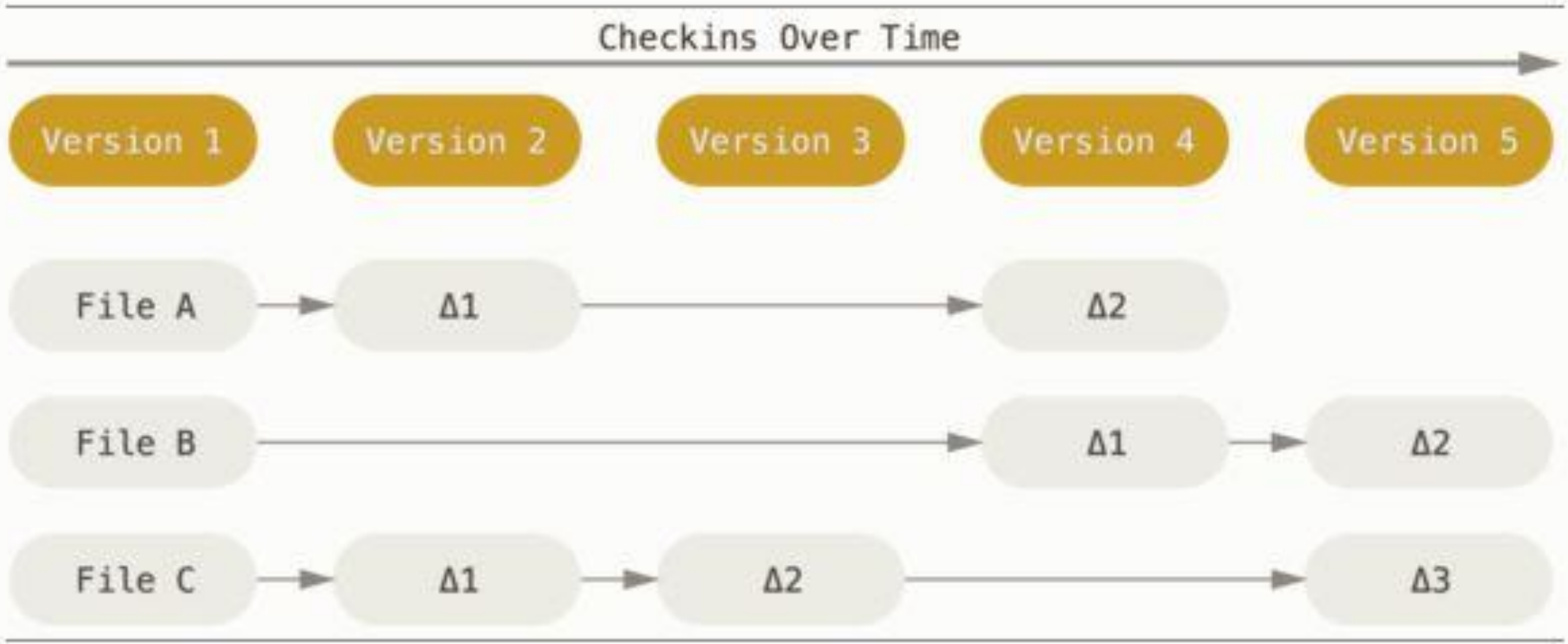

Figure 1-4. Storing data as changes to a base version of each file.

From: ProGit, Chapter 1.3 – Getting Started, link: http://git-scm.com/book/en/v2/Getting-Started-Git-Basics

Authored by Scott Chacon and Ben Straub

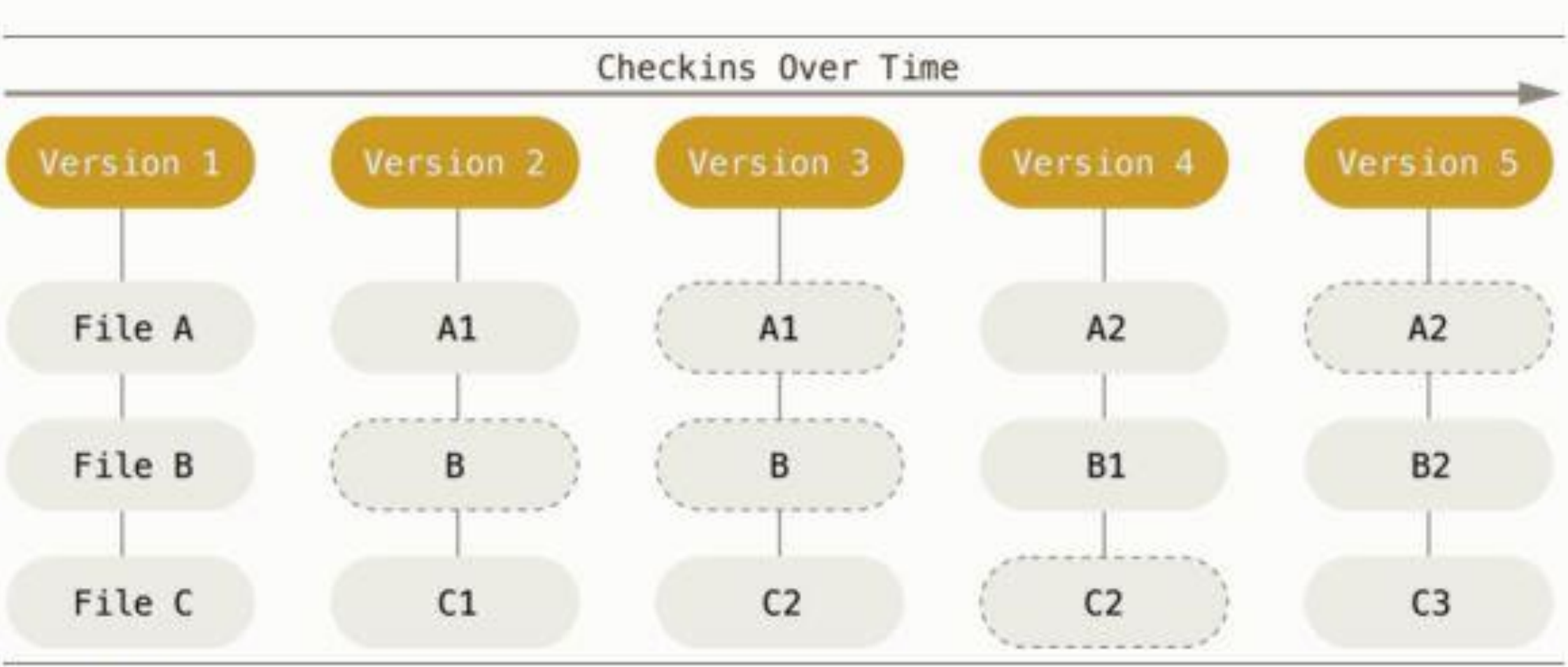

Figure 1-5. Storing data as snapshots of the project over time.

From: ProGit, Chapter 1.3 – Getting Started, link: http://git-scm.com/book/en/v2/Getting-Started-Git-Basics

Authored by Scott Chacon and Ben Straub

HOW DOES IT WORK?

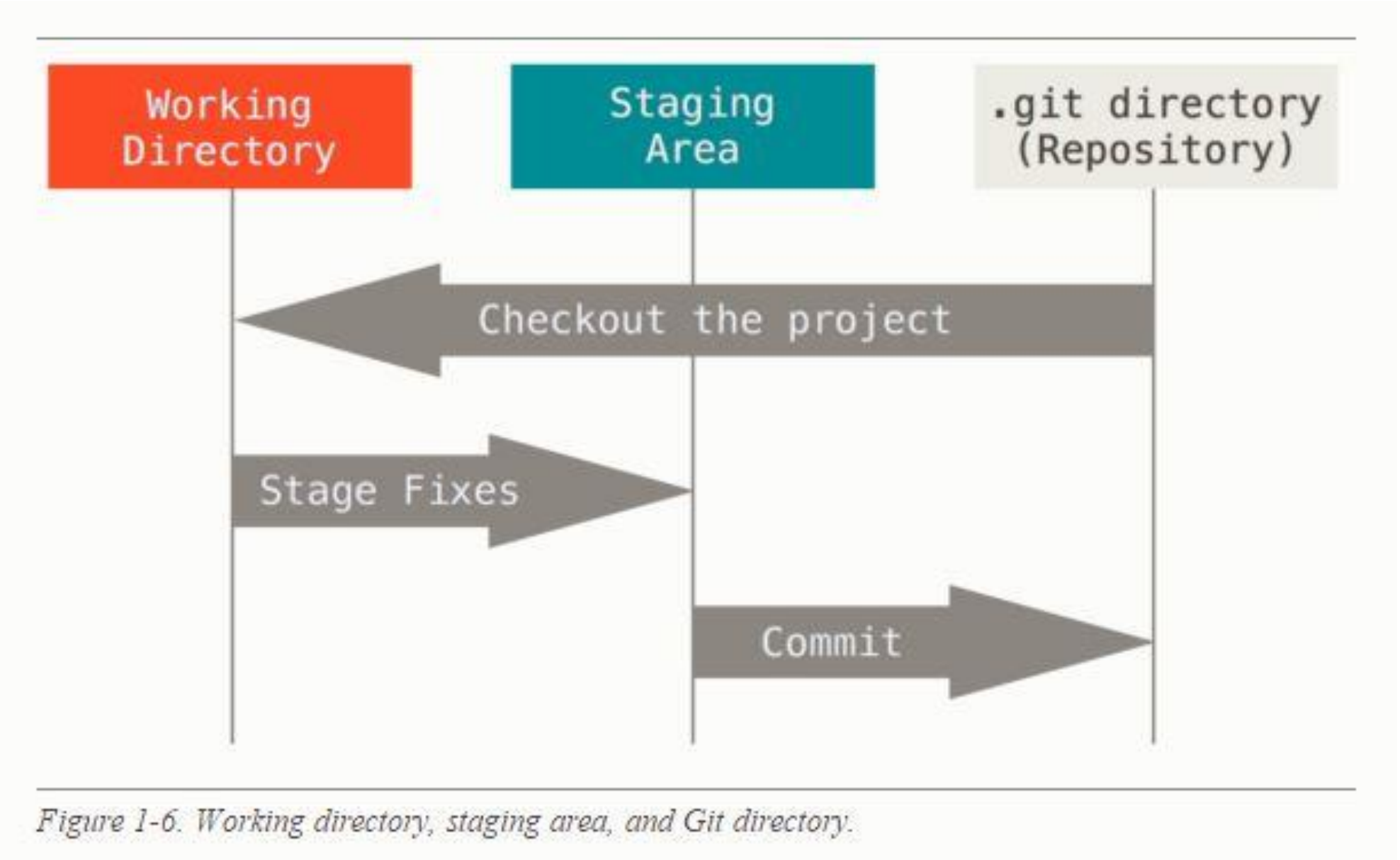

Figure 1-6. Working directory, staging area, and Git directory.

From: ProGit, Chapter 1.3 – Getting Started, link: http://git-scm.com/book/en/v2/Getting-Started-Git-Basics

Authored by Scott Chacon and Ben Straub

SOME GIT COMMANDS

Install (Linux only):

```
> sudo apt-get install git
```

Initialize a repository:

```
> git init
```

Clone a repository from a remote:

```
> git clone (URL) [optional directory]
```

Pull changes from a remote:

```
> git pull [optional remote name]
```

Track a file:

```
> git add (filename)
```

See a list of changes not tracked:

```
> git status
```

Commit changes:

```
> git commit [optional message -m “message”]
```

Push changes to a remote:

```
> git push [optional remote name]
```

COMMON REMOTE GIT REPOSITORY STORAGE

- Github.com (most common)
  - Great visual interface
  - Has a nice GUI (optional)
  - Widely regarded as the resume for a CS enthusiast
  - Public repositories are free for all users, private cost
  - Public and private repositories are free for any student (.edu email address)

We are going to use Github.com.

- Bitbucket
  - Public repositories are free for all users, private cost
  - Public and private repositories are free for any student (.edu email address)
- GitLab
  - Public and private repositories are free for students, especially for research
  - Has enterprise-level services and support, cost a decent amount
  - Fantastic data security, can pay for private server
  - Widely used in academia

LET'S SEE IT IN ACTION:

STEPS:

1. Make a github.com account, log in
2. Search for our group repository
3. Fork the repository, and store your own changes on your Github account!
4. Read the first chapter of ProGit in your free time (~30 minutes)