Drupal and Source Control Management (SCM) using Git/SVN

David Thorne

For Cambridge Drupal Users Group - March '12

Monday, 26 March 12

WARNING: Traditionally SCM is a geek heavy topic. Even if we skirt over the non git bits, it's long and complex. At best this will introduce you to the wonderful world of SCM, at worst you won't need any Nytol tonight and the IoA security team will have a shock in the morning finding you all asleep here! This is designed as an introduction to svn/git, some basic Drupal related mechanisms for git.

Contents (Or an idea of)

That said what am I going to talk about? Mostly git and SVN. I will go over basic source control ideas (System agnostic) for those new to it, but this is a git and SVN talk, therefore I will talk git/SVN. I welcome questions about other SCM's and how to use them, but I am more versed in Git than others. I used to use SVN so I am able to answer questions about that but am rusty - so some help may be appreciated from other SVN users if I come unstuck! BTW I am not an advanced git user! WHat I know barely scratches the surface of git!

Contents (Or an idea of)

- What is Source Control (Referencing Git and Subversion/SVN)

That said what am I going to talk about? Mostly git and SVN. I will go over basic source control ideas (System agnostic) for those new to it, but this is a git and SVN talk, therefore I will talk git/SVN. I welcome questions about other SCM's and how to use them, but I am more versed in Git than others. I used to use SVN so I am able to answer questions about that but am rusty - so some help may be appreciated from other SVN users if I come unstuck! BTW I am not an advanced git user! WHat I know barely scratches the surface of git!

Contents (Or an idea of)

- What is Source Control (Referencing Git and Subversion/SVN)
- Why use Source Control (Referencing Git and SVN)

That said what am I going to talk about? Mostly git and SVN. I will go over basic source control ideas (System agnostic) for those new to it, but this is a git and SVN talk, therefore I will talk git/SVN. I welcome questions about other SCM's and how to use them, but I am more versed in Git than others. I used to use SVN so I am able to answer questions about that but am rusty - so some help may be appreciated from other SVN users if I come unstuck! BTW I am not an advanced git user! WHat I know barely scratches the surface of git!

Contents (Or an idea of)

- What is Source Control (Referencing Git and Subversion/SVN)
- Why use Source Control (Referencing Git and SVN)
- Why Drupal chose Git (Summary)

That said what am I going to talk about? Mostly git and SVN. I will go over basic source control ideas (System agnostic) for those new to it, but this is a git and SVN talk, therefore I will talk git/SVN. I welcome questions about other SCM's and how to use them, but I am more versed in Git than others. I used to use SVN so I am able to answer questions about that but am rusty - so some help may be appreciated from other SVN users if I come unstuck! BTW I am not an advanced git user! WHat I know barely scratches the surface of git!

Contents (Or an idea of)

- What is Source Control (Referencing Git and Subversion/SVN)
- Why use Source Control (Referencing Git and SVN)
- Why Drupal chose Git (Summary)
- SVN basics

That said what am I going to talk about? Mostly git and SVN. I will go over basic source control ideas (System agnostic) for those new to it, but this is a git and SVN talk, therefore I will talk git/SVN. I welcome questions about other SCM's and how to use them, but I am more versed in Git than others. I used to use SVN so I am able to answer questions about that but am rusty - so some help may be appreciated from other SVN users if I come unstuck! BTW I am not an advanced git user! WHat I know barely scratches the surface of git!

Contents (Or an idea of)

- What is Source Control (Referencing Git and Subversion/SVN)
- Why use Source Control (Referencing Git and SVN)
- Why Drupal chose Git (Summary)
- SVN basics
- Git basics

That said what am I going to talk about? Mostly git and SVN. I will go over basic source control ideas (System agnostic) for those new to it, but this is a git and SVN talk, therefore I will talk git/SVN. I welcome questions about other SCM's and how to use them, but I am more versed in Git than others. I used to use SVN so I am able to answer questions about that but am rusty - so some help may be appreciated from other SVN users if I come unstuck! BTW I am not an advanced git user! WHat I know barely scratches the surface of git!

Contents (Or an idea of)

- What is Source Control (Referencing Git and Subversion/SVN)
- Why use Source Control (Referencing Git and SVN)
- Why Drupal chose Git (Summary)
- SVN basics
- Git basics
- Git workflows (Advanced topic, basic ideas)

That said what am I going to talk about? Mostly git and SVN. I will go over basic source control ideas (System agnostic) for those new to it, but this is a git and SVN talk, therefore I will talk git/SVN. I welcome questions about other SCM's and how to use them, but I am more versed in Git than others. I used to use SVN so I am able to answer questions about that but am rusty - so some help may be appreciated from other SVN users if I come unstuck! BTW I am not an advanced git user! WHat I know barely scratches the surface of git!

Contents (Or an idea of)

- What is Source Control (Referencing Git and Subversion/SVN)
- Why use Source Control (Referencing Git and SVN)
- Why Drupal chose Git (Summary)
- SVN basics
- Git basics
- Git workflows (Advanced topic, basic ideas)
- Q & A

That said what am I going to talk about? Mostly git and SVN. I will go over basic source control ideas (System agnostic) for those new to it, but this is a git and SVN talk, therefore I will talk git/SVN. I welcome questions about other SCM's and how to use them, but I am more versed in Git than others. I used to use SVN so I am able to answer questions about that but am rusty - so some help may be appreciated from other SVN users if I come unstuck! BTW I am not an advanced git user! WHat I know barely scratches the surface of git!

What is Source Control (a)

- Simply put Source Control is an easy way to help prevent screwing up when working in teams!

SCM - An easy concept but difficult to do well! Teams screw things over. Plain and simple!

What is Source Control Example 1

For example Anthony and Jeff work on a site. Jeff = Modules, Anthony = Themes. HOWEVER Jeff spots a violation of the HTML5 standard and changes it for Anthony. No SCM = overwrite. When Anthony uploads his work, all Jeff's changes are lost, kittens die, all round sadness *Sad geek face*

What is Source Control Example 1

- Jeff does modules

For example Anthony and Jeff work on a site. Jeff = Modules, Anthony = Themes. HOWEVER Jeff spots a violation of the HTML5 standard and changes it for Anthony. No SCM = overwrite. When Anthony uploads his work, all Jeff's changes are lost, kittens die, all round sadness *Sad geek face*

What is Source Control Example 1

- Jeff does modules
- Anthony does themes

For example Anthony and Jeff work on a site. Jeff = Modules, Anthony = Themes. HOWEVER Jeff spots a violation of the HTML5 standard and changes it for Anthony. No SCM = overwrite. When Anthony uploads his work, all Jeff's changes are lost, kittens die, all round sadness *Sad geek face*

What is Source Control Example 1

- Jeff does modules
- Anthony does themes
- Jeff spots violation of HTML5 and corrects it

For example Anthony and Jeff work on a site. Jeff = Modules, Anthony = Themes. HOWEVER Jeff spots a violation of the HTML5 standard and changes it for Anthony. No SCM = overwrite. When Anthony uploads his work, all Jeff's changes are lost, kittens die, all round sadness *Sad geek face*

What is Source Control Example 1

- Jeff does modules
- Anthony does themes
- Jeff spots violation of HTML5 and corrects it
- No SCM = Overwrite

For example Anthony and Jeff work on a site. Jeff = Modules, Anthony = Themes. HOWEVER Jeff spots a violation of the HTML5 standard and changes it for Anthony. No SCM = overwrite. When Anthony uploads his work, all Jeff's changes are lost, kittens die, all round sadness *Sad geek face*

:-(

Sad geek :-(

What is Source Control Example 1

- Jeff does modules
- Anthony does themes
- Jeff spots violation of HTML5 and corrects it
- No SCM = Overwrite

With SCM = Patches merges and all round happiness, kittens are born (Although please note they still die if you use the WRONG SCM - The right/wrong SCM will be explained later...)

What is Source Control Example 1

- Jeff does modules
- Anthony does themes
- Jeff spots violation of HTML5 and corrects it

With SCM = Patches merges and all round happiness, kittens are born (Although please note they still die if you use the WRONG SCM - The right/wrong SCM will be explained later...)

What is Source Control Example 1

- Jeff does modules
- Anthony does themes
- Jeff spots violation of HTML5 and corrects it

Merge FTW

What is Source Control Example 1

- Jeff does modules
- Anthony does themes
- Jeff spots violation of HTML5 and corrects it
- SCM = Merge

Merge FTW

:-)

do your little geek dance

What is Source Control

- Simply put Source Control is an easy way to help prevent screwing up when working in teams!

Form of backup - easy way to get back what you overwrote (Like that working function from 4PM on Friday when it's 8PM and site should have been live 2 hours ago...) IT SHOULD NOT HOWEVER BE CONSIDERED A FORM OF BACKUP! (Just saves on attrib's like module_name_wtf_please_f-ing_work)
It's essential - no arguments - it is!

What is Source Control

- Simply put Source Control is an easy way to help prevent screwing up when working in teams!
- It's a form of backup

Form of backup - easy way to get back what you overwrote (Like that working function from 4PM on Friday when it's 8PM and site should have been live 2 hours ago...) IT SHOULD NOT HOWEVER BE CONSIDERED A FORM OF BACKUP! (Just saves on attrib's like module_name_wtf_please_f-ing_work)
It's essential - no arguments - it is!

What is Source Control

- Simply put Source Control is an easy way to help prevent screwing up when working in teams!
- It's a form of backup
- It's essential

Form of backup - easy way to get back what you overwrote (Like that working function from 4PM on Friday when it's 8PM and site should have been live 2 hours ago...) IT SHOULD NOT HOWEVER BE CONSIDERED A FORM OF BACKUP! (Just saves on attrib's like module_name_wtf_please_f-ing_work)
It's essential - no arguments - it is!

What is Source Control (b)

2 primary types of SCM Distrib and central. Git (Linux Kernel, Drupal, PHP, important things) hg (Growl, Mozilla, Java, Vi) bzr (er yeah... ubuntu - the forgotten few *hang head in remembrance*) SVN - some public projects (Trac, Subversion itself, ?Apache?) but mostly now appears to be in use in private companies. CVS er yeah Eclipse? Although their migrating to Git!

What is Source Control (b)

- In software development there are 2 primary forms:

2 primary types of SCM Distrib and central. Git (Linux Kernel, Drupal, PHP, important things) hg (Growl, Mozilla, Java, Vi) bzr (er yeah... ubuntu - the forgotten few *hang head in remembrance*) SVN - some public projects (Trac, Subversion itself, ?Apache?) but mostly now appears to be in use in private companies. CVS er yeah Eclipse? Although their migrating to Git!

What is Source Control (b)

- In software development there are 2 primary forms:
  - Distributed (git, hg, bzr)

2 primary types of SCM Distrib and central. Git (Linux Kernel, Drupal, PHP, important things) hg (Growl, Mozilla, Java, Vi) bzr (er yeah... ubuntu - the forgotten few *hang head in remembrance*) SVN - some public projects (Trac, Subversion itself, ?Apache?) but mostly now appears to be in use in private companies. CVS er yeah Eclipse? Although their migrating to Git!

What is Source Control (b)

- In software development there are 2 primary forms:
  - Distributed (git, hg, bzr)
  - Centralised (svn, CVS)

2 primary types of SCM Distrib and central. Git (Linux Kernel, Drupal, PHP, important things) hg (Growl, Mozilla, Java, Vi) bzr (er yeah... ubuntu - the forgotten few *hang head in remembrance*) SVN - some public projects (Trac, Subversion itself, ?Apache?) but mostly now appears to be in use in private companies. CVS er yeah Eclipse? Although their migrating to Git!

What is Source Control (b)

- In software development there are 2 primary forms:
  - Distributed (git, hg, bzr)
  - Centralised (svn, CVS)
  - Other types exist but out of scope

2 primary types of SCM Distrib and central. Git (Linux Kernel, Drupal, PHP, important things) hg (Growl, Mozilla, Java, Vi) bzr (er yeah... ubuntu - the forgotten few *hang head in remembrance*) SVN - some public projects (Trac, Subversion itself, ?Apache?) but mostly now appears to be in use in private companies. CVS er yeah Eclipse? Although their migrating to Git!

Why Git

The last point says it all!

Why Git

- Why not?

The last point says it all!

Why Git

- Why not?
- It's distributed

The last point says it all!

Why Git

- Why not?
- It's distributed
- It's Linus' beast

The last point says it all!

Why Git

- Why not?
- It's distributed
- It's Linus' beast
- GITHUB!

The last point says it all!

Why Git

- Why not?
- It's distributed
- It's Linus' beast
- GITHUB!
- branches

The last point says it all!

Why Git

- Why not?
- It's distributed
- It's Linus' beast
- GITHUB!
- branches
- Drupal uses it

The last point says it all!

Why Git (b)

complicated - so called 2 stage commit (git add . && git commit -m “message” ;; git fetch {remote}/{branch} ;; git merge {remote/branch} {branch} etc)
Linus’ attitude very much “I’m smart you’re not - live with it”

Why Git (b)

- It's over complicated

complicated - so called 2 stage commit (git add . && git commit -m "message" ;; git fetch {remote}/{branch} ;; git merge {remote/branch} {branch} etc)
Linus' attitude very much "I'm smart you're not - live with it"

Why Git (b)

- It's over complicated
- It's Linus' beast (I'm smart - you're not attitude)

complicated - so called 2 stage commit (git add . && git commit -m "message" ;; git fetch {remote}/{branch} ;; git merge {remote/branch} {branch} etc)
Linus' attitude very much "I'm smart you're not - live with it"

Why Git (b)

- It's over complicated
- It's Linus' beast (I'm smart - you're not attitude)
- Steep learning curve (See above!)

complicated - so called 2 stage commit (git add . && git commit -m "message" ;; git fetch {remote}/{branch} ;; git merge {remote/branch} {branch} etc)
Linus' attitude very much "I'm smart you're not - live with it"

Why Git (c)

Why Git (c)

- Linus' reasoning (Everything else is stupid)

Why Git (c)

- Linus' reasoning (Everything else is stupid)
  - http://youtu.be/4XpnKHJAok8 (An hour long!)

Why Drupal chose Git

OK I knew people were going to ask this so I stuck a few slides in - but it really was very long, very complex. In fact if I remember rightly Jeff was a regular commenter on the issue!

Why Drupal chose Git

- It was git vs Bzr

OK I knew people were going to ask this so I stuck a few slides in - but it really was very long, very complex. In fact if I remember rightly Jeff was a regular commenter on the issue!

Why Drupal chose Git

- It was git vs Bzr
- http://groups.drupal.org/node/48818?page=2#comment-133893 (summary)

OK I knew people were going to ask this so I stuck a few slides in - but it really was very long, very complex. In fact if I remember rightly Jeff was a regular commenter on the issue!

Why Drupal chose Git

- It was git vs Bzr
- http://groups.drupal.org/node/48818?page=2#comment-133893 (summary)
  - coders were leaving for github

OK I knew people were going to ask this so I stuck a few slides in - but it really was very long, very complex. In fact if I remember rightly Jeff was a regular commenter on the issue!

Why Drupal chose Git

- It was git vs Bzr
- http://groups.drupal.org/node/48818?page=2#comment-133893 (summary)
  - coders were leaving for github
  - larger community

OK I knew people were going to ask this so I stuck a few slides in - but it really was very long, very complex. In fact if I remember rightly Jeff was a regular commenter on the issue!

SVN Basics

Tags say it all

SVN Basics

- svn checkout {url} (svn co)

Tags say it all

SVN Basics

- svn checkout {url} (svn co)
- svn add {file}

Tags say it all

SVN Basics

- svn checkout {url} (svn co)
- svn add {file}
- svn commit (svn ci)

Tags say it all

SVN Basics

- svn checkout {url} (svn co)
- svn add {file}
- svn commit (svn ci)
- svn delete (svn del svn rm)

Tags say it all

SVN Basics

- svn checkout {url} (svn co)
- svn add {file}
- svn commit (svn ci)
- svn delete (svn del svn rm)
- svn tag

Tags say it all

SVN Basics

- svn checkout {url} (svn co)
- svn add {file}
- svn commit (svn ci)
- svn delete (svn del svn rm)
- svn tag
- svn branch

Tags say it all

Git Basics

explain tags - explain add/rm needs to be before EVERY commit. Explain there are other ways of doing it but beyond scope of project. Explain what a remote is - give example of origin, but further explanation is out of scope

Git Basics

- git clone {url}

explain tags - explain add/rm needs to be before EVERY commit. Explain there are other ways of doing it but beyond scope of project. Explain what a remote is - give example of origin, but further explanation is out of scope

Git Basics

- git clone {url}
- git add {file list} // git rm {file list} (DIFFERENT MEANING)

explain tags - explain add/rm needs to be before EVERY commit. Explain there are other ways of doing it but beyond scope of project. Explain what a remote is - give example of origin, but further explanation is out of scope

Git Basics

- git clone {url}
- git add {file list} // git rm {file list} (DIFFERENT MEANING)
- git commit {file list}

explain tags - explain add/rm needs to be before EVERY commit. Explain there are other ways of doing it but beyond scope of project. Explain what a remote is - give example of origin, but further explanation is out of scope

Git Basics

- git clone {url}
- git add {file list} // git rm {file list} (DIFFERENT MEANING)
- git commit {file list}
- git push {remote} {branch}:{branch}

explain tags - explain add/rm needs to be before EVERY commit. Explain there are other ways of doing it but beyond scope of project. Explain what a remote is - give example of origin, but further explanation is out of scope

Git Basics

- git clone {url}
- git add {file list} // git rm {file list} (DIFFERENT MEANING)
- git commit {file list}
- git push {remote} {branch}:{branch}
- git pull {remote} {branch}:{branch}

explain tags - explain add/rm needs to be before EVERY commit. Explain there are other ways of doing it but beyond scope of project. Explain what a remote is - give example of origin, but further explanation is out of scope

Git Basics

- git clone {url}
- git add {file list} // git rm {file list} (DIFFERENT MEANING)
- git commit {file list}
- git push {remote} {branch}:{branch}
- git pull {remote} {branch}:{branch}
- git branch {branch name} (Or git checkout -b {branch} {commit})

explain tags - explain add/rm needs to be before EVERY commit. Explain there are other ways of doing it but beyond scope of project. Explain what a remote is - give example of origin, but further explanation is out of scope

Git Basics

- git clone {url}
- git add {file list} // git rm {file list} (DIFFERENT MEANING)
- git commit {file list}
- git push {remote} {branch}:{branch}
- git pull {remote} {branch}:{branch}
- git branch {branch name} (Or git checkout -b {branch} {commit})
- git tag

explain tags - explain add/rm needs to be before EVERY commit. Explain there are other ways of doing it but beyond scope of project. Explain what a remote is - give example of origin, but further explanation is out of scope

Git Workflows

Online ebook by Scott Chacon. Jeff - forgive me here - I am minimising how Drupal uses Git, but it's about as simple as I can get!

Git Workflows

- Look at http://progit.org (Online ebook)

Online ebook by Scott Chacon. Jeff - forgive me here - I am minimising how Drupal uses Git, but it's about as simple as I can get!

Git Workflows

- Look at http://progit.org (Online ebook)
- Centralised - AKA "The one ring" (SVN, to a degree)

Online ebook by Scott Chacon. Jeff - forgive me here - I am minimising how Drupal uses Git, but it's about as simple as I can get!

Git Workflows - Centralised

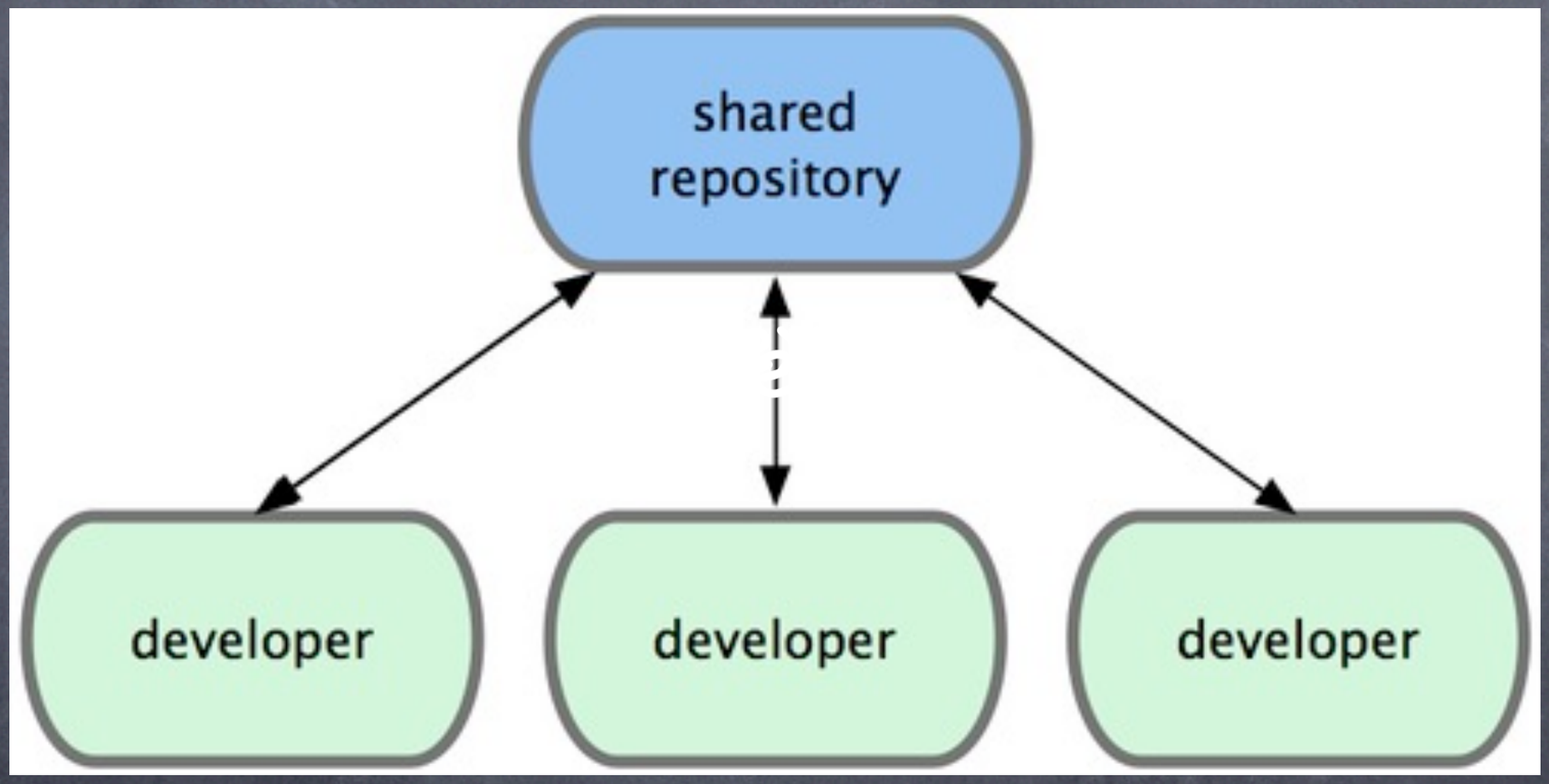

Photo credit: http://progit.org/book/ch5-1.html CC NC Sharealike 3 license

Git Workflows

- Look at http://progit.org (Online ebook)
- Centralised - AKA "The one ring" (SVN, to a degree)

Online ebook by Scott Chacon. Jeff - forgive me here - I am minimising how Drupal uses Git, but it's about as simple as I can get!

Git Workflows

- Look at http://progit.org (Online ebook)
- Centralised - AKA "The one ring" (SVN, to a degree)
- Integration Manager (Blessed, Drupal to a degree)

Online ebook by Scott Chacon. Jeff - forgive me here - I am minimising how Drupal uses Git, but it's about as simple as I can get!

Git Workflows - Integration

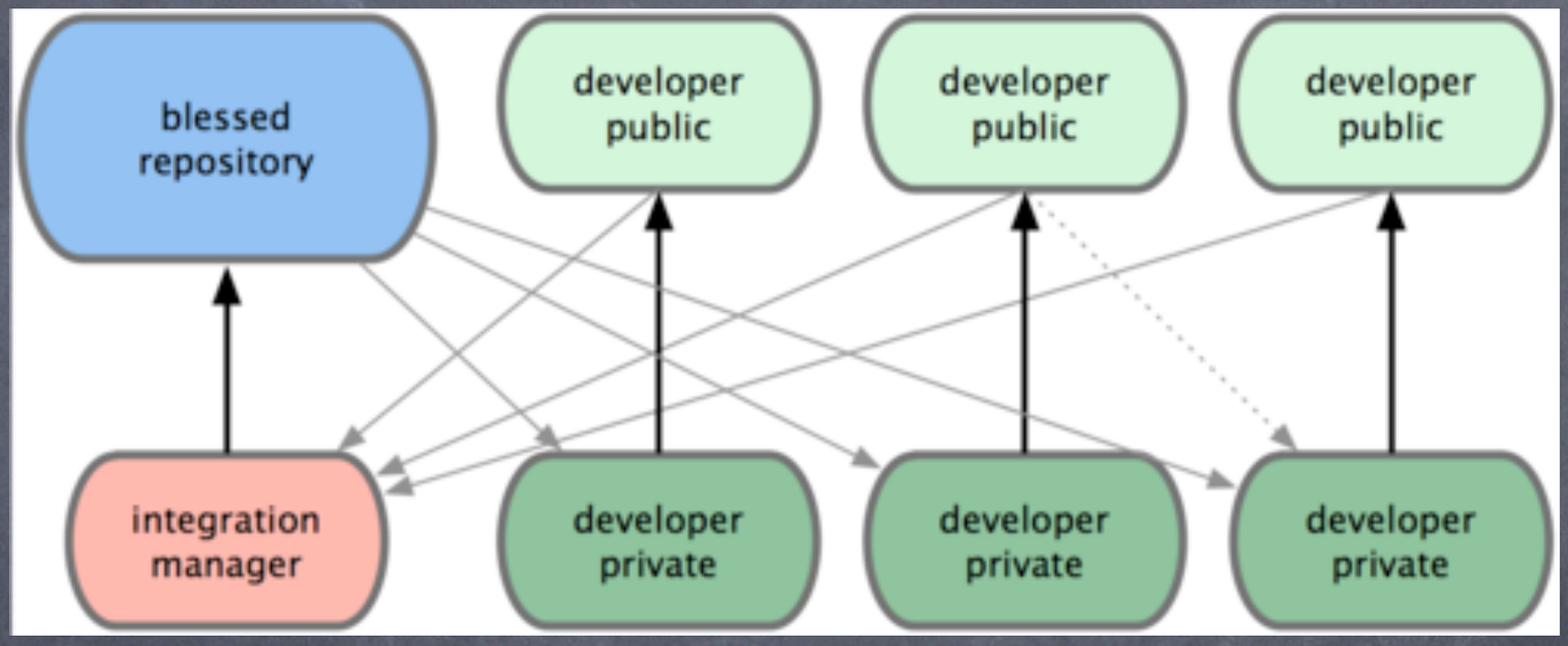

Photo credit: http://progit.org/book/ch5-1.html CC NC Sharealike 3 license

This is a bit like a lot of Drupal modules, if you imagine the integration manager as the maintainer/co-maintainer. So the integration manager takes the patches, and commits them to the main repo.

Git Workflows

- Look at http://progit.org (Online ebook)
- Centralised - AKA "The one ring" (SVN, to a degree)
- Integration Manager (Blessed, Drupal to a degree)

Online ebook by Scott Chacon. Jeff - forgive me here - I am minimising how Drupal uses Git, but it's about as simple as I can get!

Git Workflows

- Look at http://progit.org (Online ebook)
- Centralised - AKA "The one ring" (SVN, to a degree)
- Integration Manager (Blessed, Drupal to a degree)
- Dictator and Leuitennant (Blessed with few key commiters - Drupal Core, to a degree)

Online ebook by Scott Chacon. Jeff - forgive me here - I am minimising how Drupal uses Git, but it's about as simple as I can get!

Git Workflows - BDFL

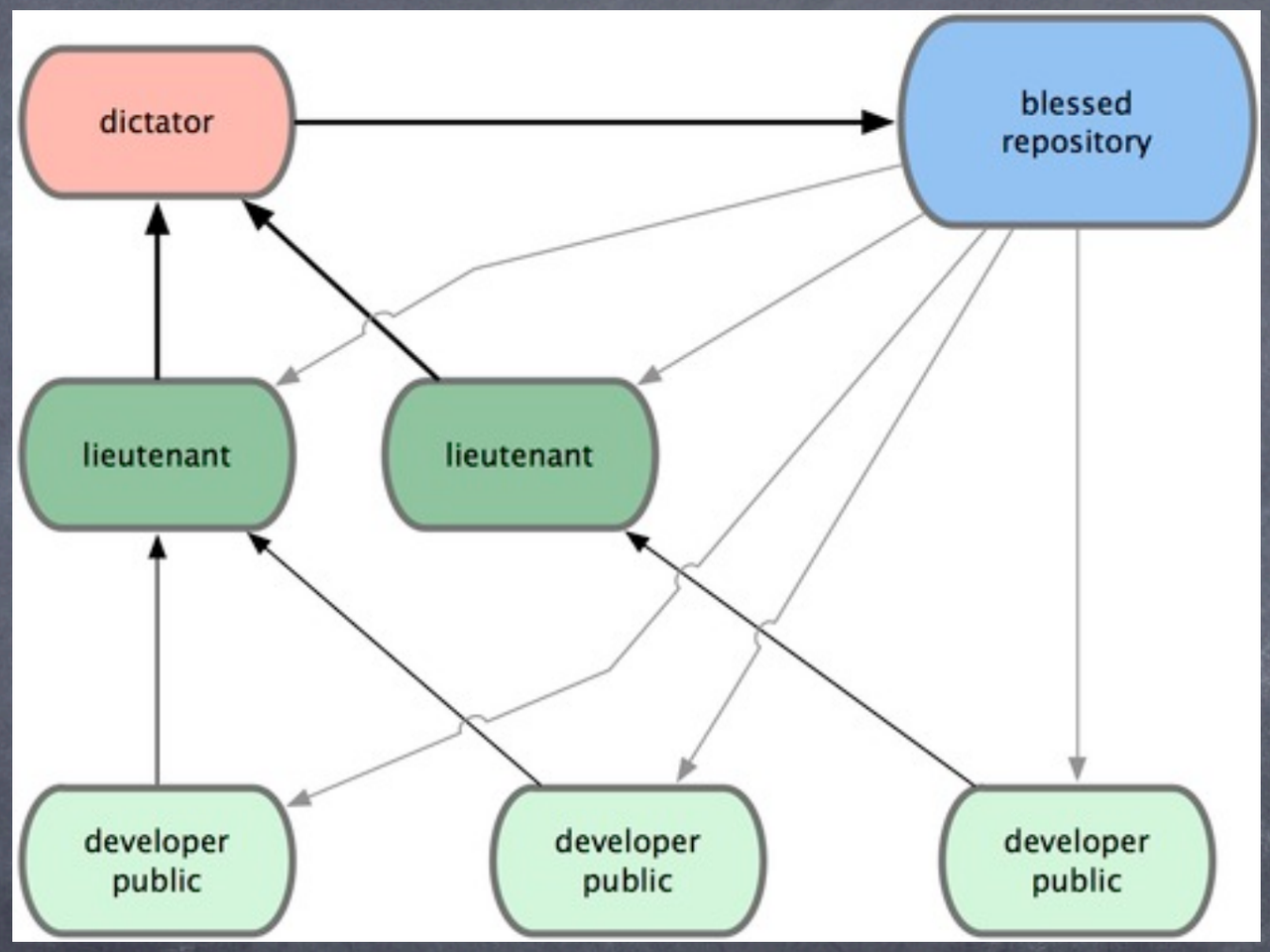

Photo credit: http://progit.org/book/ch5-1.html CC NC Sharealike 3 license

This is a bit like D8, With lieutenants being the Inititive leaders (Gábor Hojtsy, Kris “EclipseGC” Vanderwater, etc) and the dictator being Dries, or Nat “Catch” Catchpole

Q & A

Q & A

- Over to you...