
ODTUG
Kscope16
CHICAGO, ILLINOIS · JUNE 26-30
PLEASE FILL OUT YOUR EVALUATIONS

my. HARVARD

Agile Oracle BI Development for Multiple Users with Git

Yes, it can be done

Harvard University

Founded 1636
20,000 active students
7,500 degrees awarded/year
2,475 faculty
18,000 total employees

Eric Brown

Eric is a Senior Business Intelligence Engineer on Harvard University's new Student Information Systems/BI Apps implementation.

In addition to ETL development and day-to-day administrative tasks, he in charge of the development lifecycle and release process for all OBIEE and ETL work.

He's been with Harvard since 2014 and holds a degree from Dartmouth College in pure mathematics.

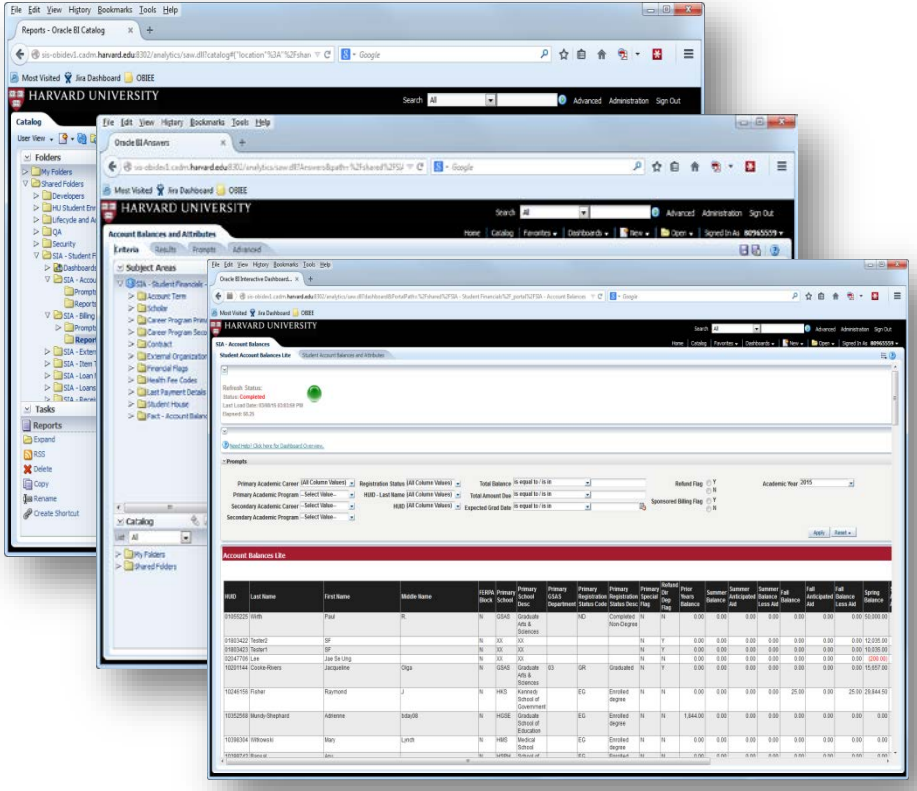
His favorite food is pizza.

Agenda

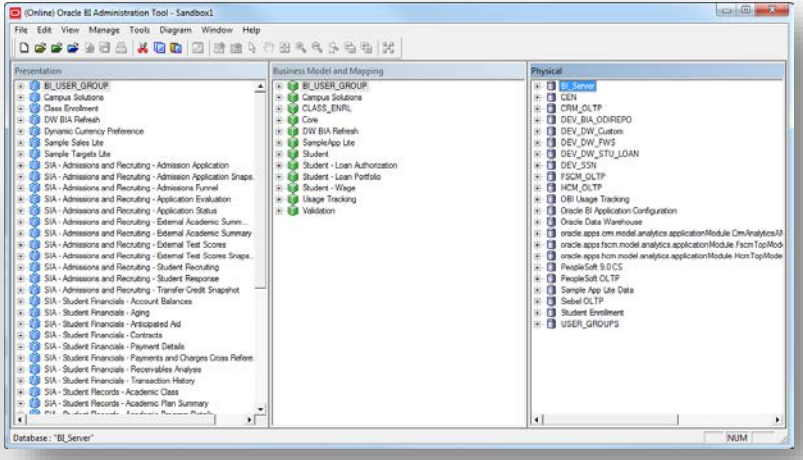
- Where We Were
- Where We Wanted to Be
- How We Got There: Overview
- The Process in Action
- How We Got There: Detail
- Where We Are Going from Here

OBIEE Development Domains

Catalog



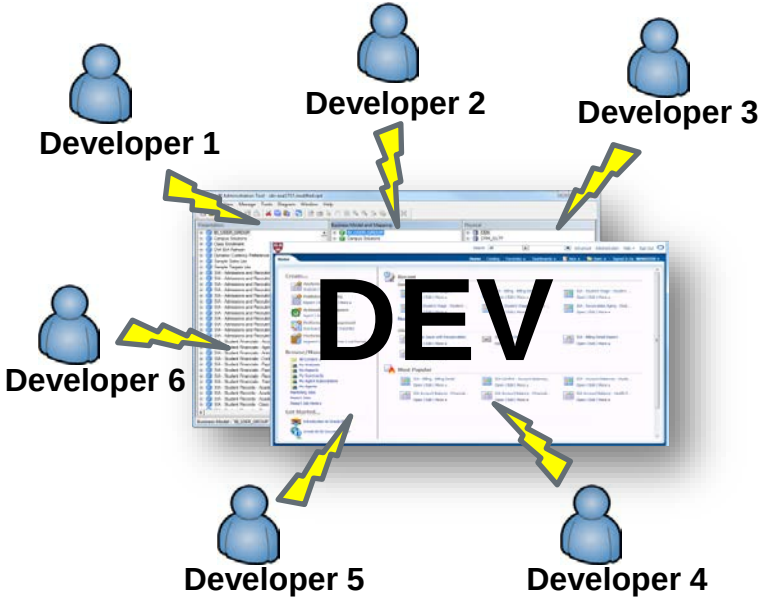
RPD



Where We Were

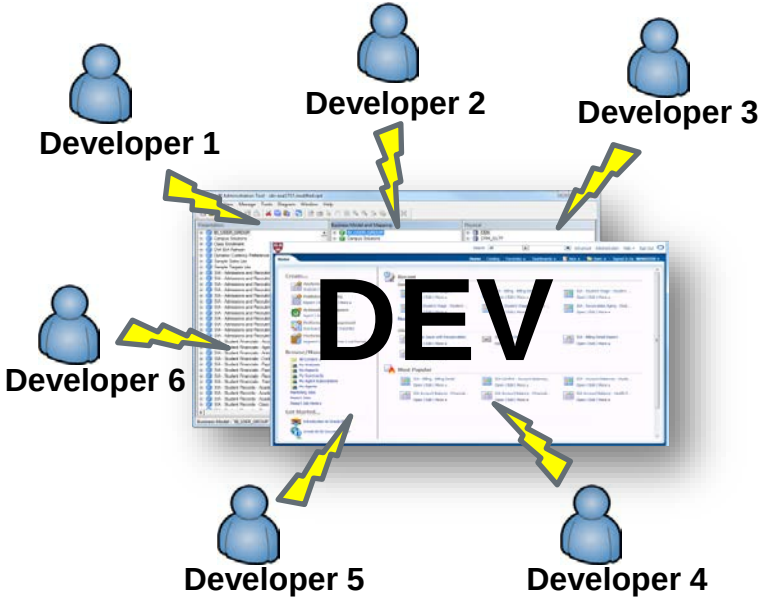
Where We Were

- Single DEV environment
- Concurrent online development by many developers
- Catalog development straightforward but...



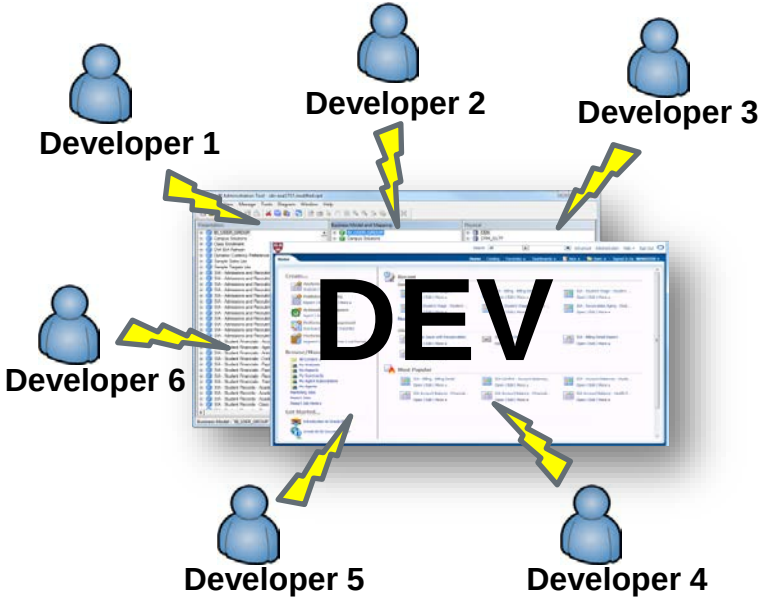
Where We Were

- Single DEV environment
- Concurrent online development by many developers
- Catalog development straightforward but...
- Difficult to coordinate RPD work



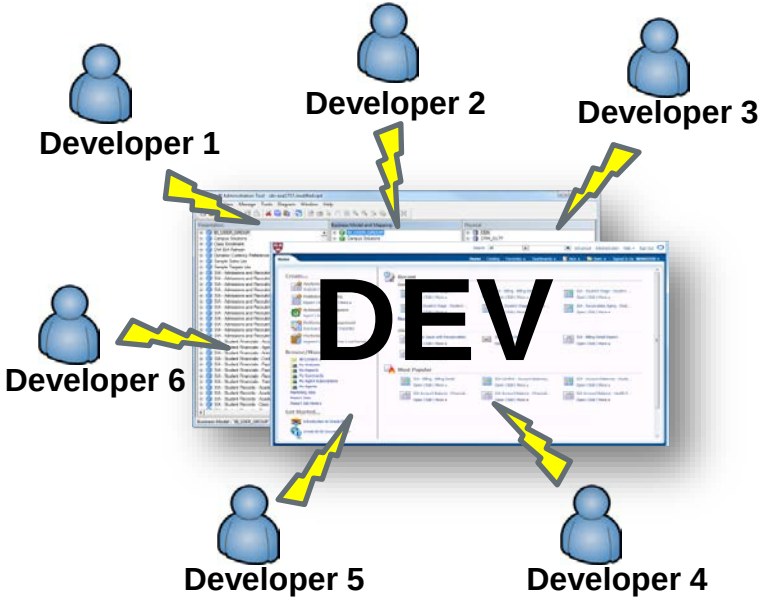
Where We Were

- Single DEV environment
- Concurrent online development by many developers
- Catalog development straightforward but...
- Difficult to coordinate RPD work
- Disappearing development



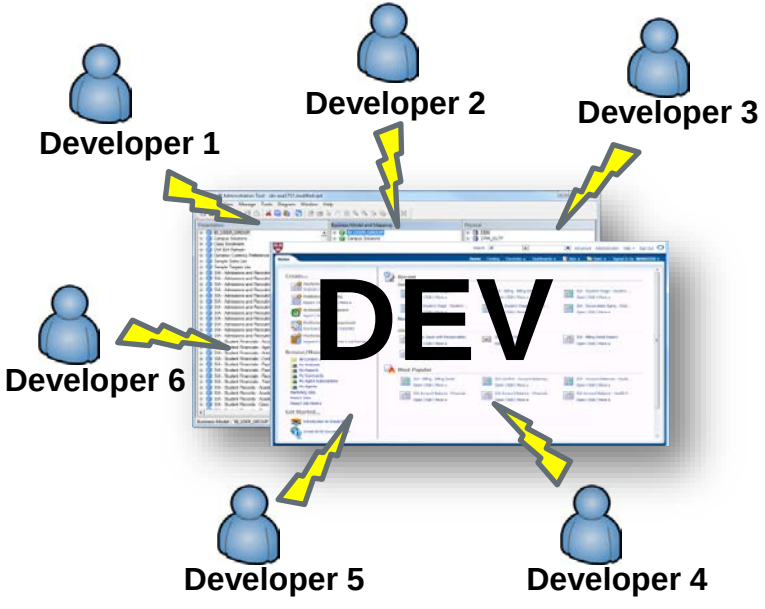
Where We Were

- Single DEV environment
- Concurrent online development by many developers
- Catalog development straightforward but...
- Difficult to coordinate RPD work
- Disappearing development
- Fear of trying new things



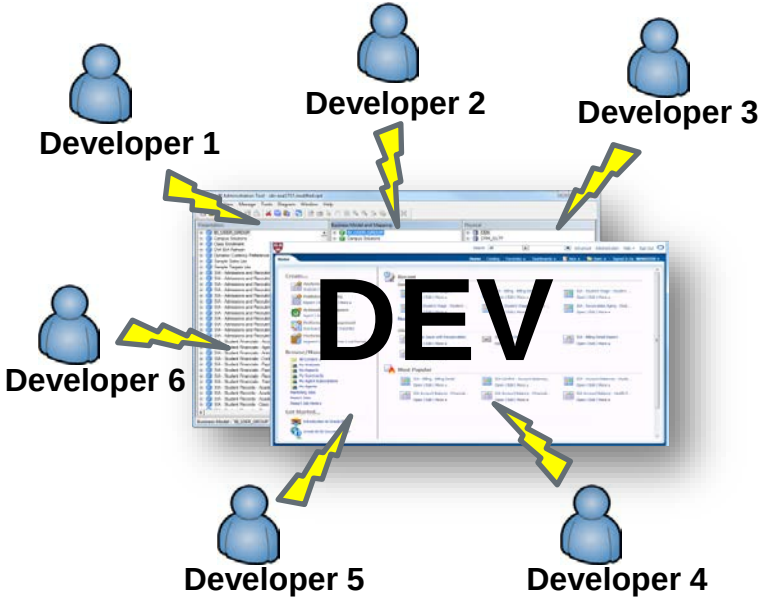
Where We Were

- Single DEV environment
- Concurrent online development by many developers
- Catalog development straightforward but...
- Difficult to coordinate RPD work
- Disappearing development
- Fear of trying new things
- Impossible to audit changes



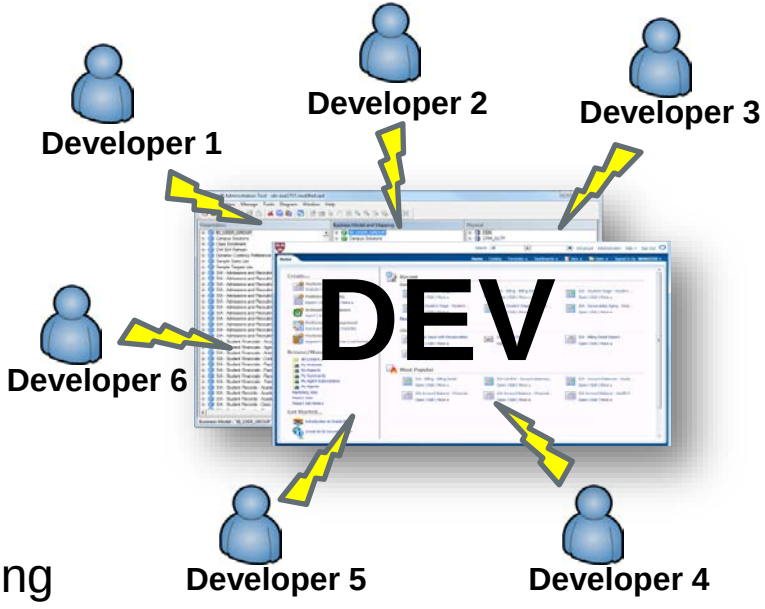
Where We Were

- Single DEV environment
- Concurrent online development by many developers
- Catalog development straightforward but...
- Difficult to coordinate RPD work
- Disappearing development
- Fear of trying new things
- Impossible to audit changes
- Merging with the Admin tool is a nightmare



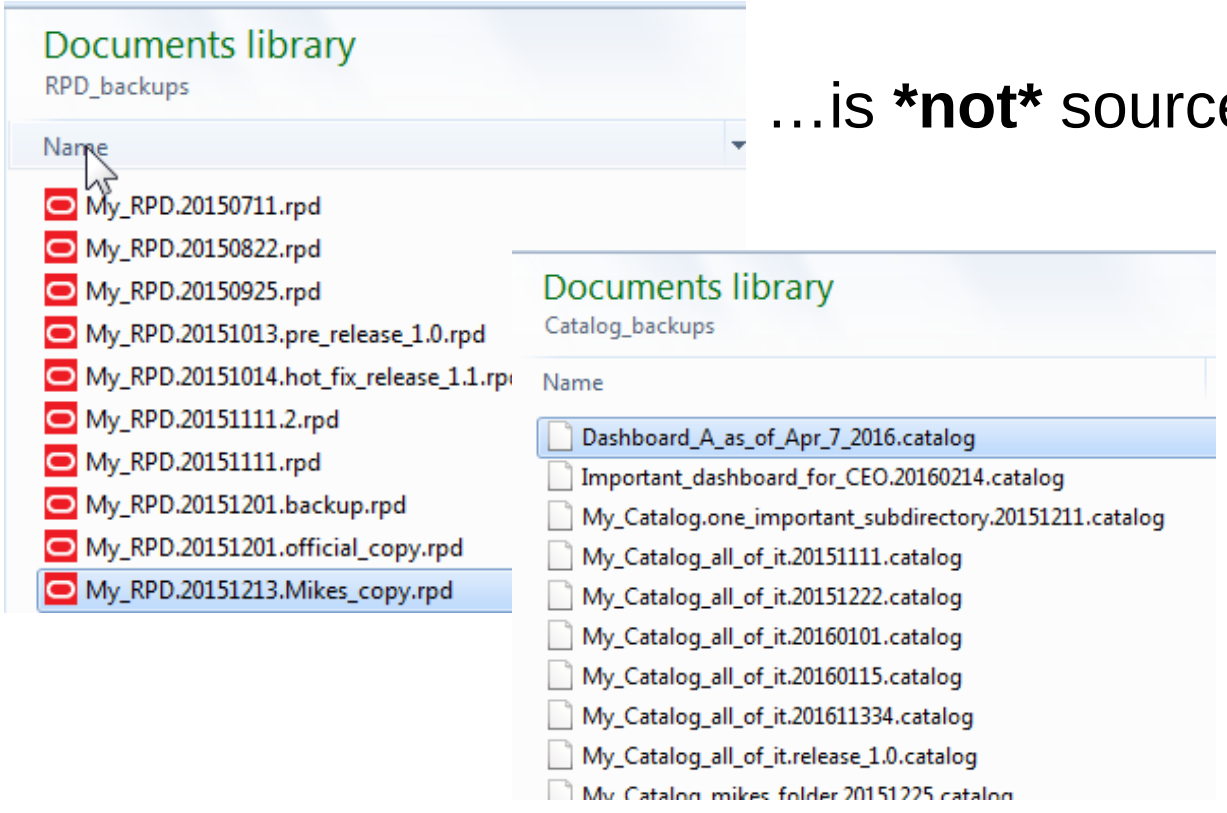
Where We Were

- Single DEV environment
- Concurrent online development by many developers
- Catalog development straightforward but...
- Difficult to coordinate RPD work
- Disappearing development
- Fear of trying new things
- Impossible to audit changes
- Merging with the Admin tool is a nightmare
- Un-Agile—promoting changes is all-or-nothing



Where We Were

This…



…is ***not*** source control.

Where We Wanted to Be

Where We Wanted to Be

S

- Simple
  - The development process must not get in the way of development

Where We Wanted to Be

SA

- Simple
  - The development process must not get in the way of development
- Agile
  - We’re an Agile shop

Where We Wanted to Be

SAS

- Simple
  - The development process must not get in the way of development
- Agile
  - We’re an Agile shop
- Safe
  - Developers must feel safe to experiment

Where We Wanted to Be

SAS

- Simple
  - The development process must not get in the way of development
- Agile
  - We’re an Agile shop
- Safe
  - Developers must feel safe to experiment
  - Development must be safe, i.e. it doesn’t mystically disappear

Where We Wanted to Be

SASA

- Simple
  - The development process must not get in the way of development
- Agile
  - We’re an Agile shop
- Safe
  - Developers must feel safe to experiment
  - Development must be safe, i.e. it doesn’t mystically disappear
- Auditable
  - Must be able to track who, what, when & why for each change

How We Got There: Overview

How We Got There: Overview

I’ve got two two words for you:

Version Control

Sand Boxes

How We Got There: Overview

About Version Control

- “Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.”*
- Works well for text files; not so well for binaries

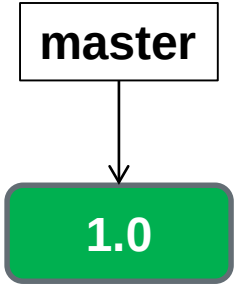
About Git

- Git is a version control system created by Linus Torvalds to maintain the Linux kernel
- It has many fans
- Why Git? lmgtfy.com/?q=why+git
- Vocabulary:
  - branch – a version, or line of development
  - commit – a save point along a branch
  - merge – to bring two or more branches together

***from *Pro Git* by Scott Chacon and Ben Straub**

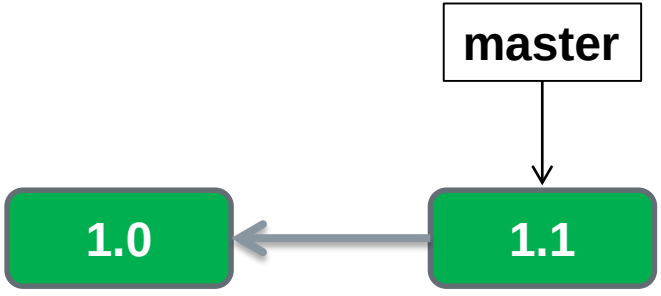
How We Got There: Overview

Branching and Merging in a Nutshell



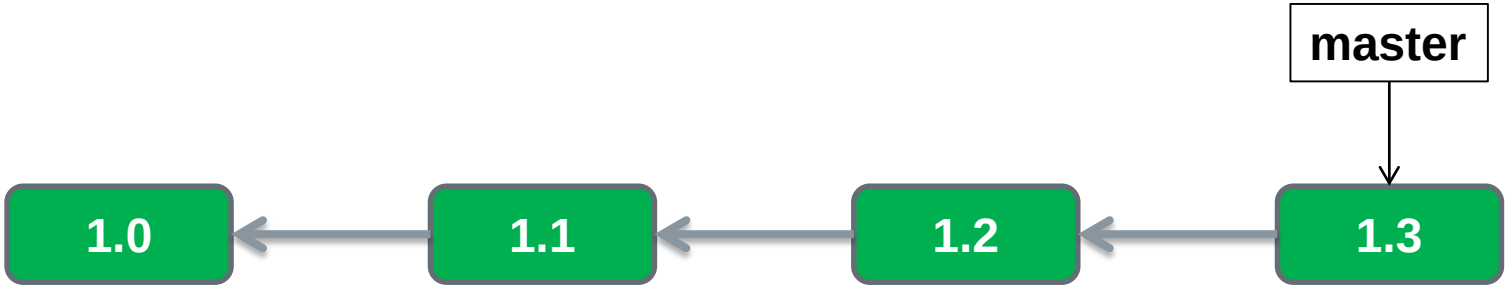
How We Got There: Overview

Branching and Merging in a Nutshell



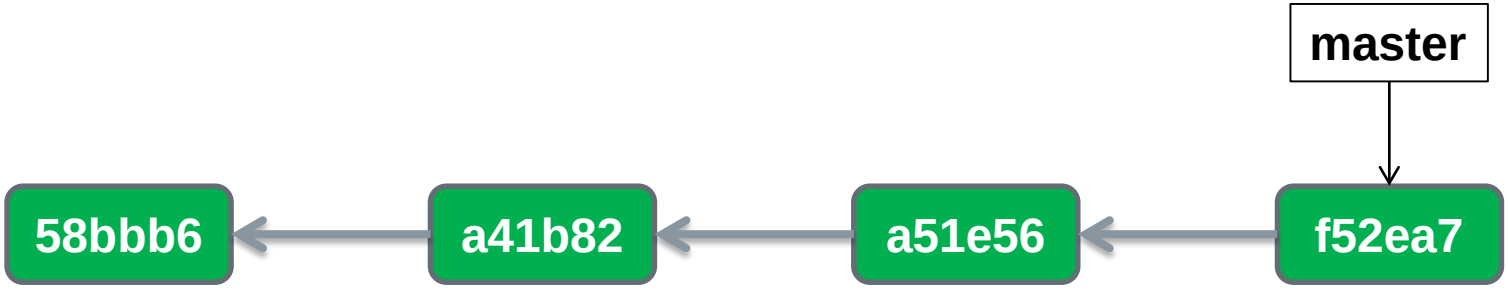
How We Got There: Overview

Branching and Merging in a Nutshell



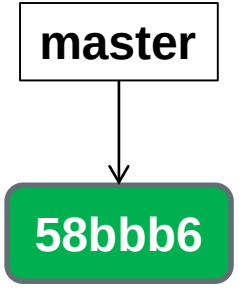
How We Got There: Overview

Branching and Merging in a Nutshell



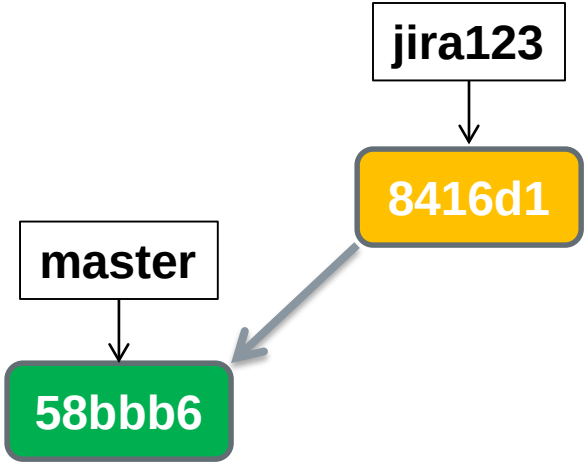
How We Got There: Overview

Branching and Merging in a Nutshell



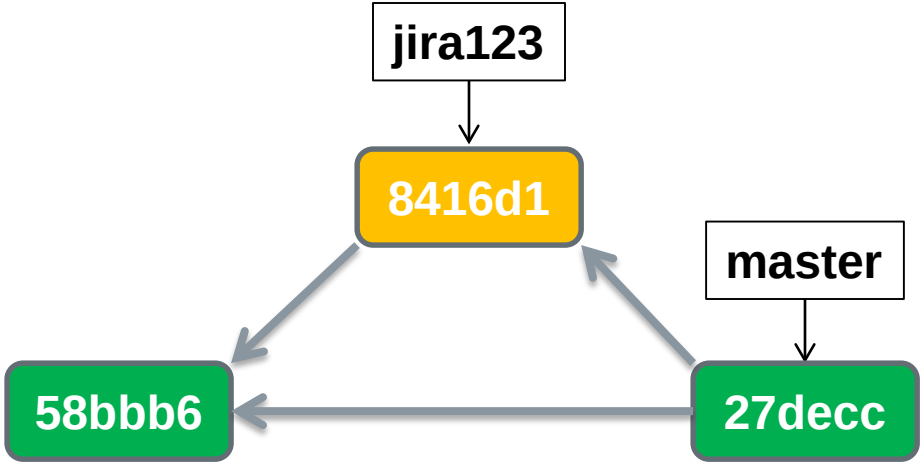
How We Got There: Overview

Branching and Merging in a Nutshell



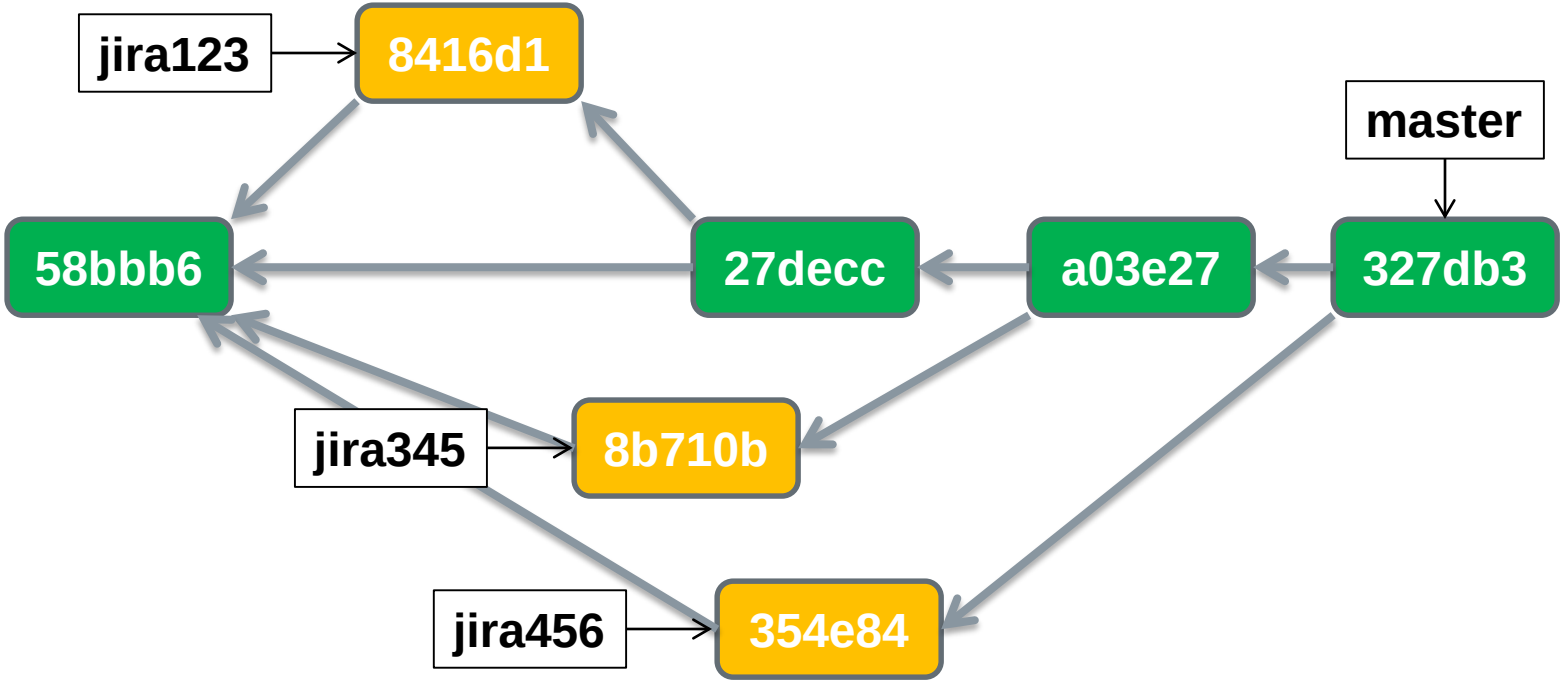
How We Got There: Overview

Branching and Merging in a Nutshell



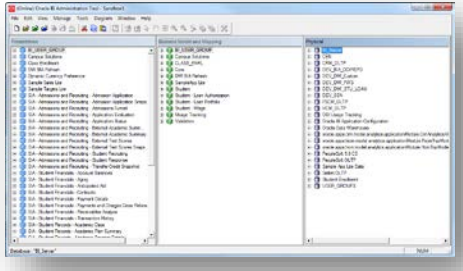
How We Got There: Overview

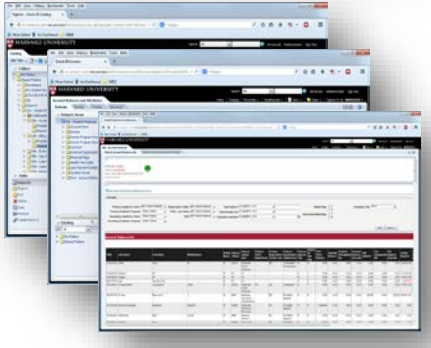
Branching and Merging in a Nutshell



How We Got There: Overview

Ok so how do we apply this to OBIEE?





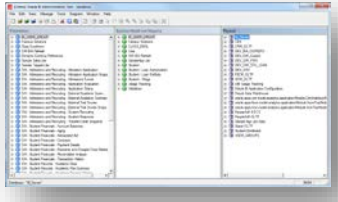
?

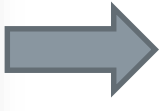
git

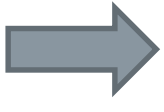
How We Got There: Overview

- Fortunately, the catalog consists primarily of text files so we can essentially just go ahead and put the catalog, or a least parts of the catalog, in Git

- Unfortunately, the RPD doesn't, so it has to be converted to XML first







How We Got There: Overview

Now that we’ve got that figured out…

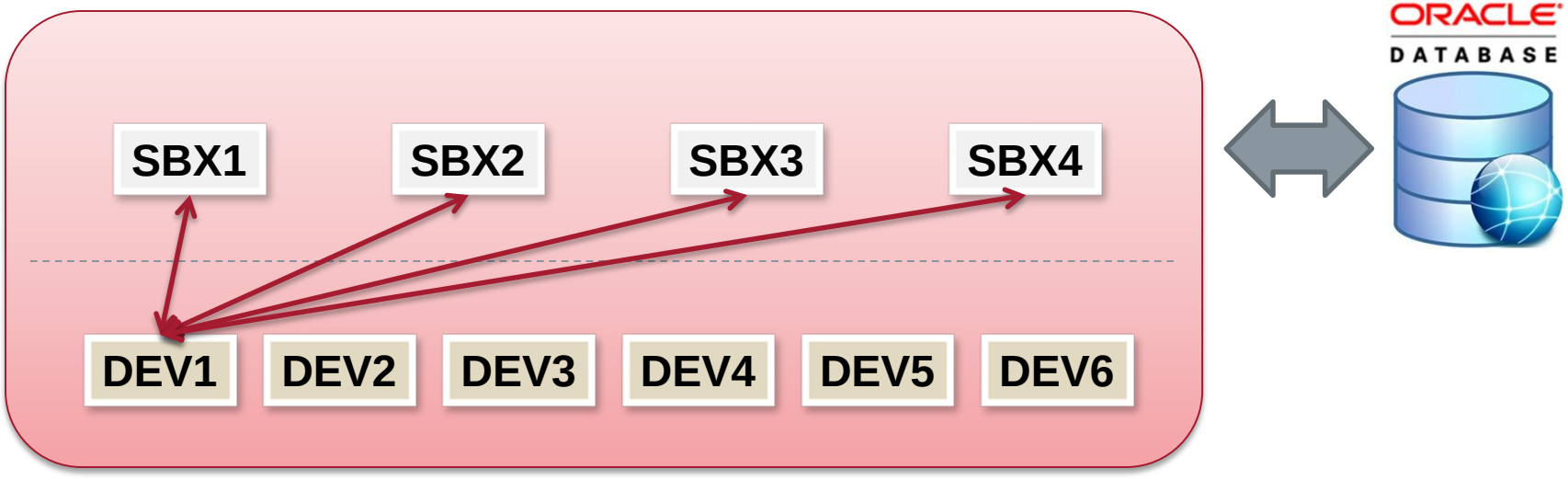
Lets talk Sandboxes

Many of the pain points we had were due to concurrent online development by multiple developers, or CODBMD

To resolve them each developer *must* have her own play space

How We Got There: Overview

Sandbox Rig



- Each SBXn is a distinct OBIEE instance
- All SBX point to a shared data warehouse
- Each DEVn is a developer's Git repository with RPD and Catalog
- A developer can deploy to any SBX

The Process in Action

The Development Process in Action

Development Lifecycle

1. Develop
2. Peer Review
3. Merge to master
4. PO/Manager Review
5. Release to Test
6. Release to Prod

The Development Process in Action

Development Lifecycle

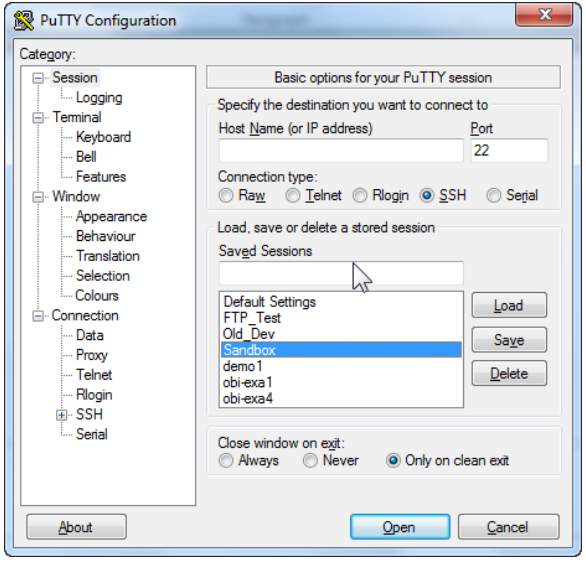
1. **Develop**
2. Peer Review
3. Merge to master
4. PO/Manager Review
5. Release to Test
6. Release to Prod

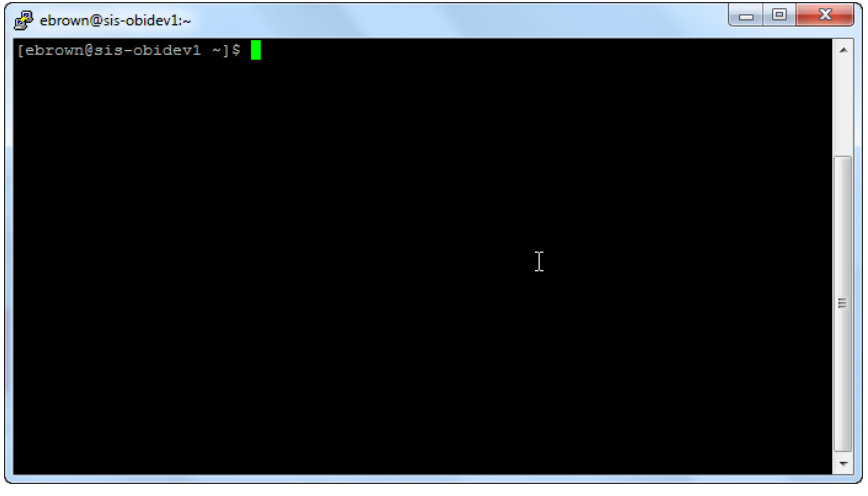
1. Log in to server
2. Checkout or create branch
3. Deploy
4. Develop
5. Pre-commit
6. Commit

The Development Process in Action

Development Process Steps

1. Log in to server

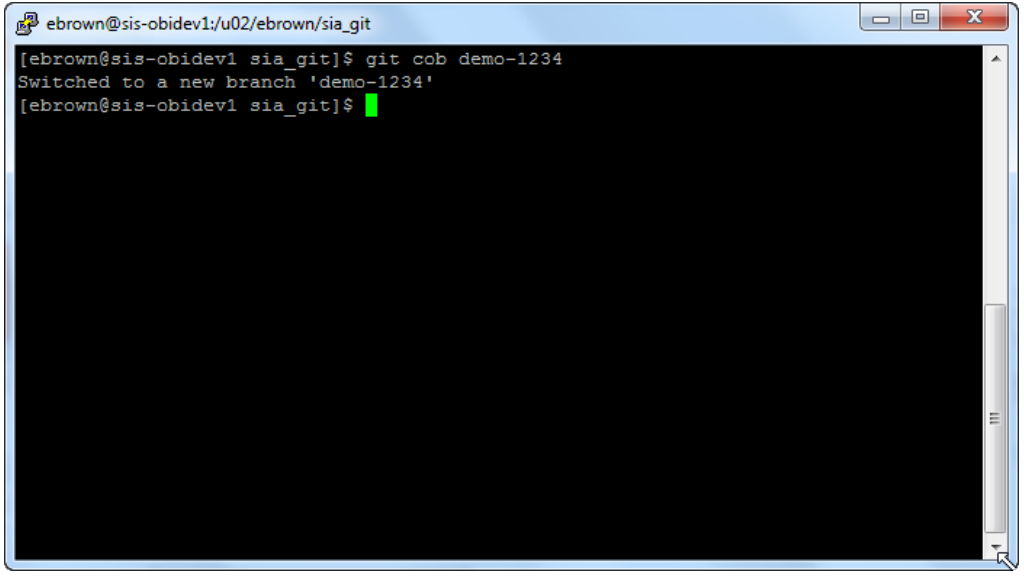




The Development Process in Action

Development Process Steps

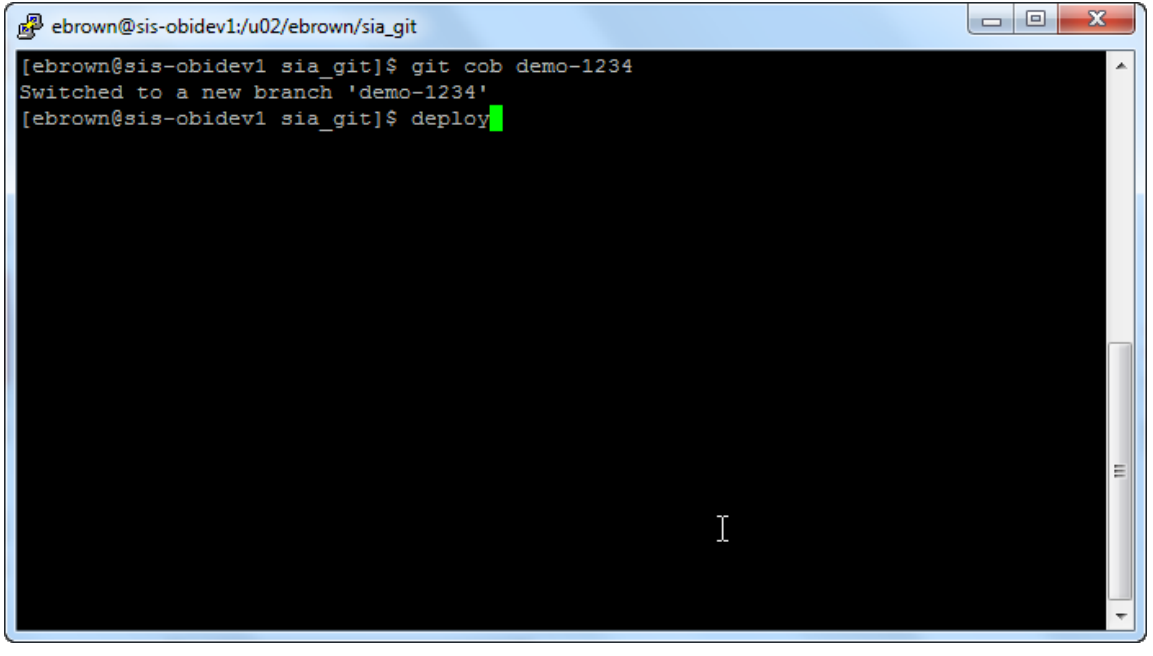
2. Checkout or create branch



The Development Process in Action

Development Process Steps

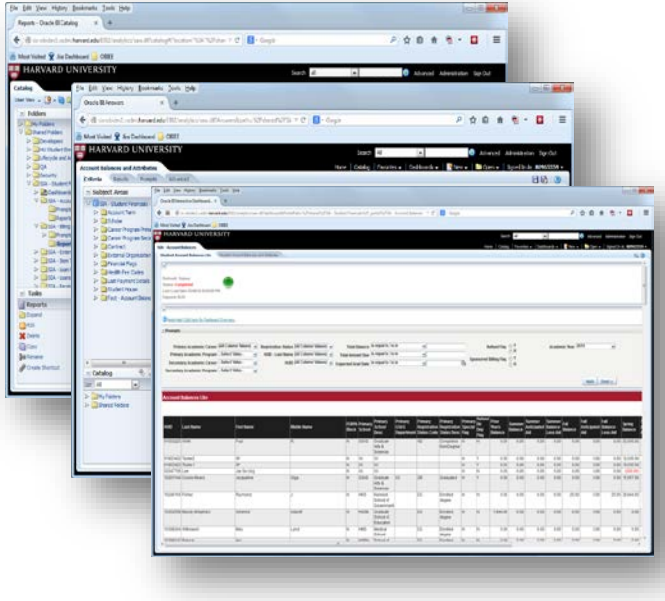
3. Deploy

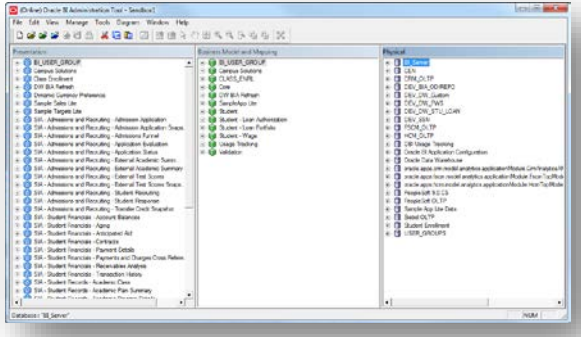


The Development Process in Action

Development Process Steps

4. Develop

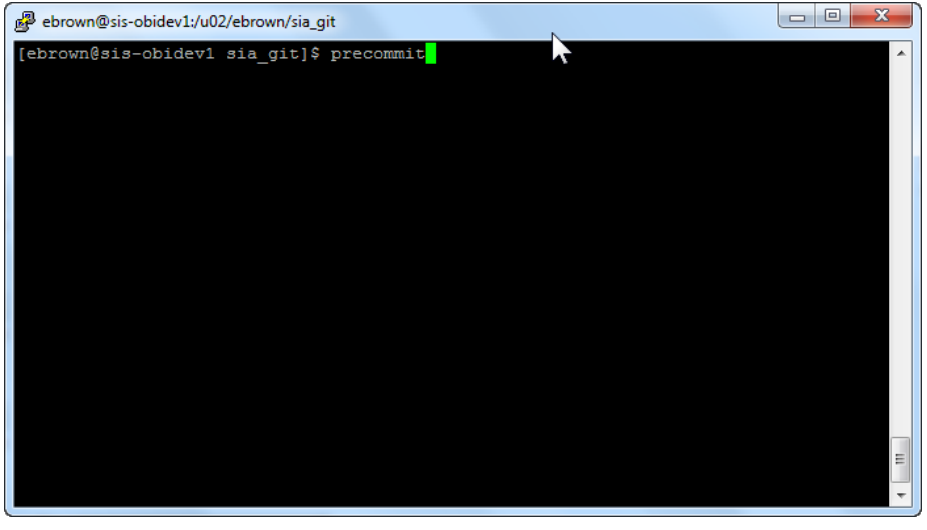




The Development Process in Action

Development Process Steps

5. Pre-commit



The Development Process in Action

Development Process Steps

6. Commit

The Development Process in Action

In Short:

1. Login, branch, deploy
2. Develop
3. Pre-commit, commit

#FTW

The Development Process in Action

Let’s do a practical example together.

The Development Process in Action

Development Lifecycle

1. **Develop**
2. Peer Review
3. Merge to master
4. PO/Manager Review
5. Release to Test
6. Release to Prod

1. Log in to server
2. Checkout or create branch
3. Deploy
4. Develop
5. Pre-commit
6. Commit

The Development Process in Action

Development Lifecycle

1. Develop
2. **Peer Review**
3. Merge to master
4. PO/Manager Review
5. Release to Test
6. Release to Prod

1. Log in to server
2. Checkout branch
3. Deploy
4. Review
5. Approve/reject in Jira

The Development Process in Action

Development Lifecycle

1. Develop
2. Peer Review
3. **Merge to master**
4. PO/Manager Review
5. Release to Test
6. Release to Prod

1. Log in to server
2. git merge
3. git push

#FTW

The Development Process in Action

Development Lifecycle

1. Develop
2. Peer Review
3. Merge to master
4. **PO/Manager Review**
5. Release to Test
6. Release to Prod

1. Log in to OBIEE Dev
2. Review
3. Approve/reject in Jira

The Development Process in Action

Development Lifecycle

1. Develop
2. Peer Review
3. Merge to master
4. PO/Manager Review
5. **Release to Test**
6. Release to Prod

1. Log in to server
2. git pull
3. Deploy

The Development Process in Action

Development Lifecycle

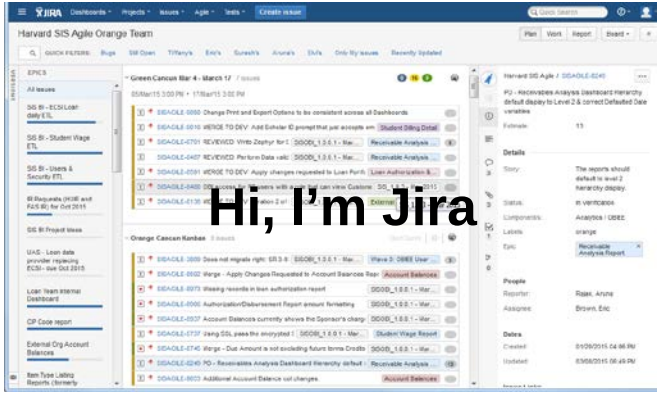
1. Develop
2. Peer Review
3. Merge to master
4. PO/Manager Review
5. Release to Test
6. **Release to Prod**

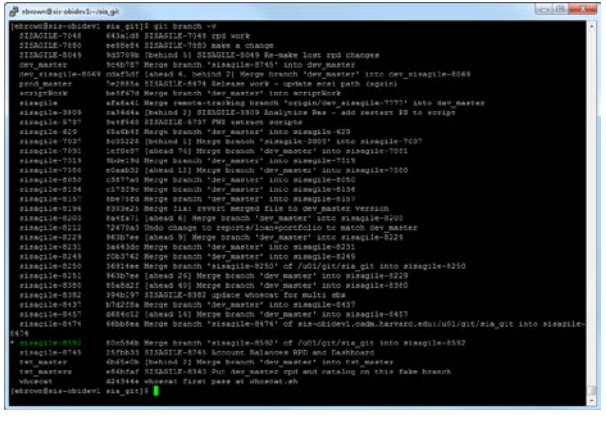
1. Log in to server
2. git pull
3. Deploy

The Development Process in Action

Is it Agile?

- Development is incremental
- Facilitates code review/feedback
- Branches organize development with Jira stories
- Release is a non-event

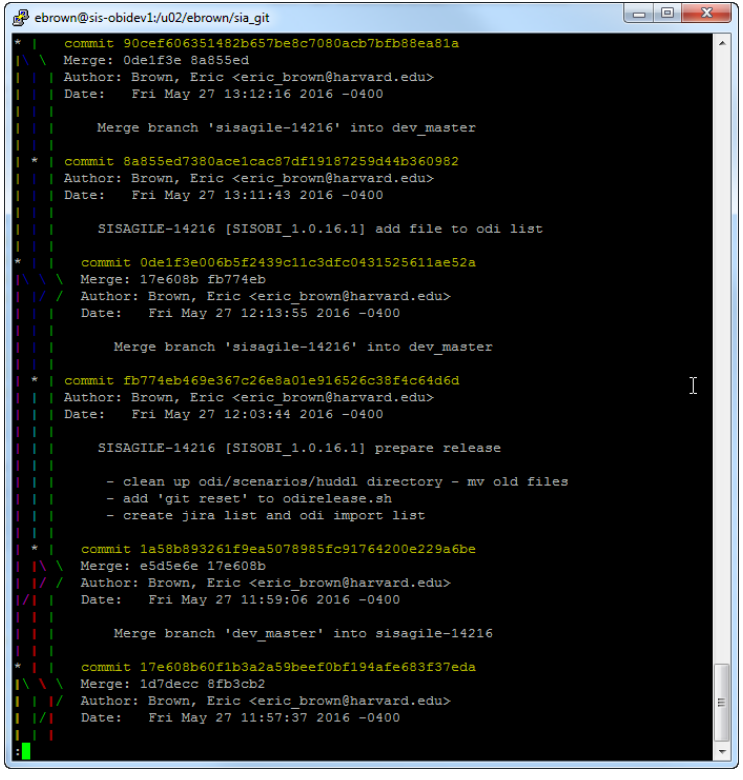




The Development Process in Action

Is it Auditable?
Yes, absolutely.

Every single change to the code base is tagged with date, developer, ticket number, comment.
Every single change can be inspected.



How We Got There: Detail

How We Got There: Detail

The magic is in the scripts

- deploy
- precommit

…and in a bit of configuration.

How We Got There: Detail

Configuration needed

Steps:

1. Copy three catalog directories Git repo
   - root/shared
   - root/system/metadata
   - root/system/privs
2. Replace these directories in MW Home with links to Git
3. Set environment variables (e.g. path to Git repo)

How We Got There: Detail

deploy

Steps:

1. Convert XML to RPD using biserverxmlexec
2. Point catalog links to my Git repo*
3. Deploy the RPD using WLST

precommit

Steps:

1. Convert RPD to XML using biserverxmlgen
2. Update catalog file permissions and ownership*
3. Add changes to Git staging area

***only needed on sandbox**

How We Got There: Detail

OK, yes there's a bit more

1. Git
2. Utilities: ogstatus, clearreorders, rxq, …
3. MY_LOGO
4. ODI…
5. Release process

Where We Are Going from Here

Where We Are Going from Here

Better utilities to examine diffs
Bitbucket
Port to Windows (*maybe*)
Automate environment configuration

Questions


ODTUG
Kscope16
CHICAGO, ILLINOIS · JUNE 26-30
PLEASE FILL OUT YOUR EVALUATIONS