

ODTUG

Kscope16



CHICAGO, ILLINOIS · JUNE 26-30

PLEASE FILL OUT YOUR EVALUATIONS



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 is 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

The screenshot displays the Oracle BI Catalog interface. The top navigation bar includes 'Reports - Oracle BI Catalog'. The main content area is titled 'Account Balances and Allocations' and shows a 'Student Account Balances' report. Below the report title, there are several filters and a data table. The table has columns for RUID, Last Name, First Name, Middle Name, F2006 Primary School, Primary School, Primary SAS, History, Primary, Primary, Student, Summer, Summer, Full, Full, and SPTD. The table contains several rows of data, including names like Paul, Robert, and James.

RUID	Last Name	First Name	Middle Name	F2006 Primary School	Primary School	Primary SAS	History	Primary	Primary	Student	Summer	Summer	Full	Full	SPTD
0195222	Walt	Paul		01016	Graduate High School	01016	Non-Career	01016	01016	01016	01016	01016	01016	01016	01016
0195242	Tobert	Robert		01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016
0204750	Lee	James		01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016
1024158	Fisher	Raymond	J	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016
1025258	Handy-Sherpherd	Andrew	Walter	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016
1028324	Worobets	May	Lynch	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016
1028823	Brace	William		01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016	01016

The screenshot displays the Oracle BI Administration Tool - Sandbox1 interface. The main content area is titled 'Business Model and Mapping' and shows a list of objects. The objects are organized into three columns: Presentation, Business Model and Mapping, and Physical. The objects include various data sources and mappings, such as BI_USER_GROUP, Campus Solutions, and Student - Loan Portfolio.

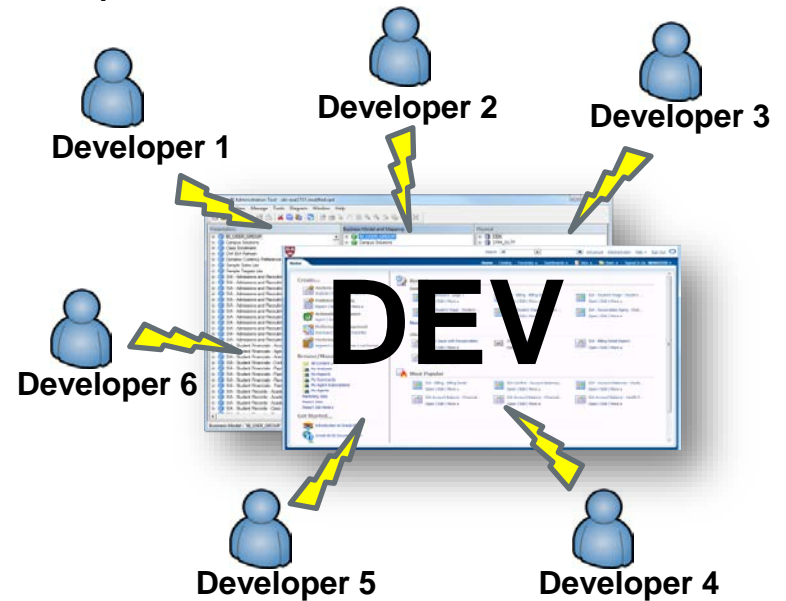
Presentation	Business Model and Mapping	Physical
BI_USER_GROUP	BI_USER_GROUP	BI_Server
Campus Solutions	Campus Solutions	CEN
Class Enrollment	CLASS_ENRL	CRN_CLTLP
DW BIA Refresh	Core	DEV_BIA_OCIROPO
Dynamic Currency Preference	DW BIA Refresh	DEV_DW_Custom
Sample Sales Lite	SampleApp Lite	DEV_DW_FWS
Sample Targets Lite	Student - Loan Authorization	DEV_DW_STU_LOAN
SIA - Admissions and Recruiting - Admission Application	Student - Loan Portfolio	DEV_SSN
SIA - Admissions and Recruiting - Admission Application Snap	Student - Loan Portfolio	FSOIN_CLTLP
SIA - Admissions and Recruiting - Admissions Funnel	Student - Wage	HCM_CLTLP
SIA - Admissions and Recruiting - Application Evaluation	Usage Tracking	OB Usage Tracking
SIA - Admissions and Recruiting - Application Status	Validation	Oracle BI Application Configuration
SIA - Admissions and Recruiting - External Academic Summ...		Oracle Data Warehouse
SIA - Admissions and Recruiting - External Academic Summary		oracle apps em model analytics applicationModule CmsAnalytics10
SIA - Admissions and Recruiting - External Test Scores		oracle apps em model analytics applicationModule HCM TopMode
SIA - Admissions and Recruiting - External Test Scores Snap		oracle apps em model analytics applicationModule HCM TopMode
SIA - Admissions and Recruiting - Student Recruiting		PeopleSoft SICS
SIA - Admissions and Recruiting - Student Response		PeopleSoft CLTLP
SIA - Admissions and Recruiting - Transfer Credit Snapshot		Sample App Use Data
SIA - Student Financials - Account Balances		Siebel CLTLP
SIA - Student Financials - Aging		Student Enrollment
SIA - Student Financials - Anticipated Aid		USER_GROUPS
SIA - Student Financials - Contracts		
SIA - Student Financials - Payment Details		
SIA - Student Financials - Payments and Charges Cross Refere		
SIA - Student Financials - Receivables Analysis		
SIA - Student Financials - Transaction History		
SIA - Student Records - Academic Class		
SIA - Student Records - Academic Plan Summary		

Where We Were



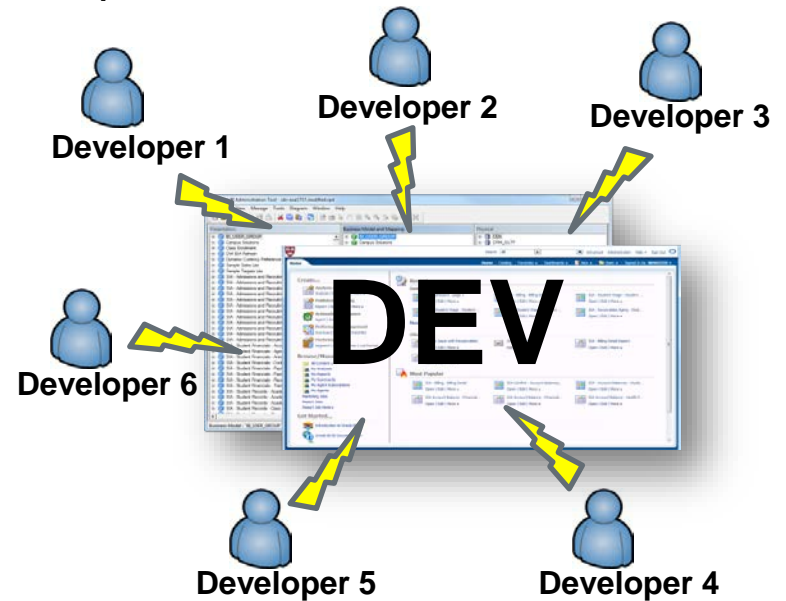
Where We Were

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



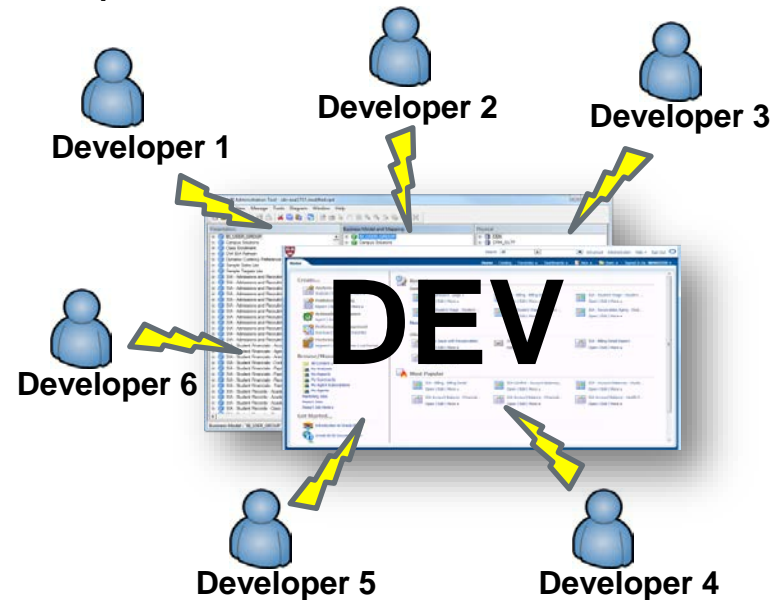
Where We Were

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



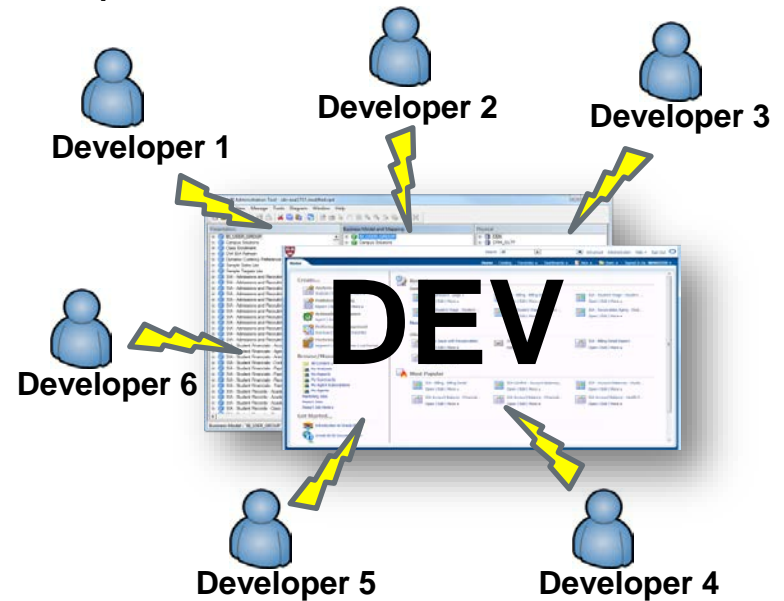
Where We Were

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



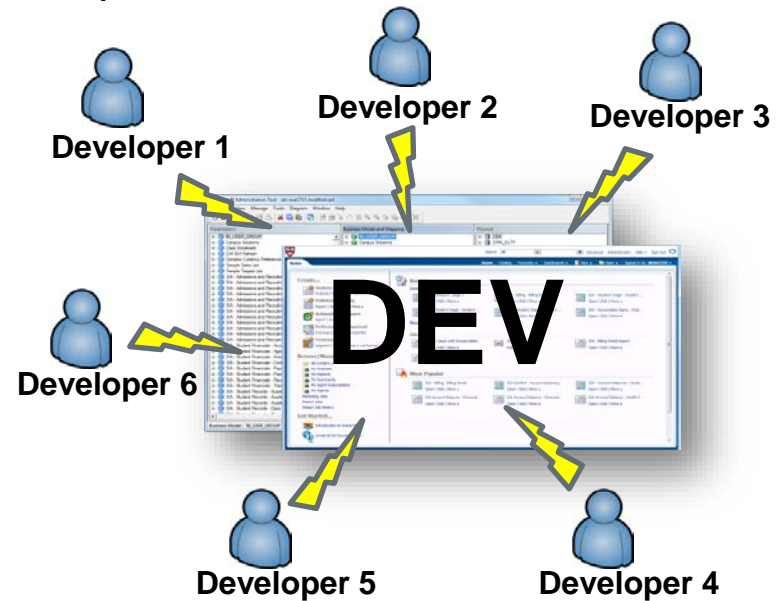
Where We Were

- 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



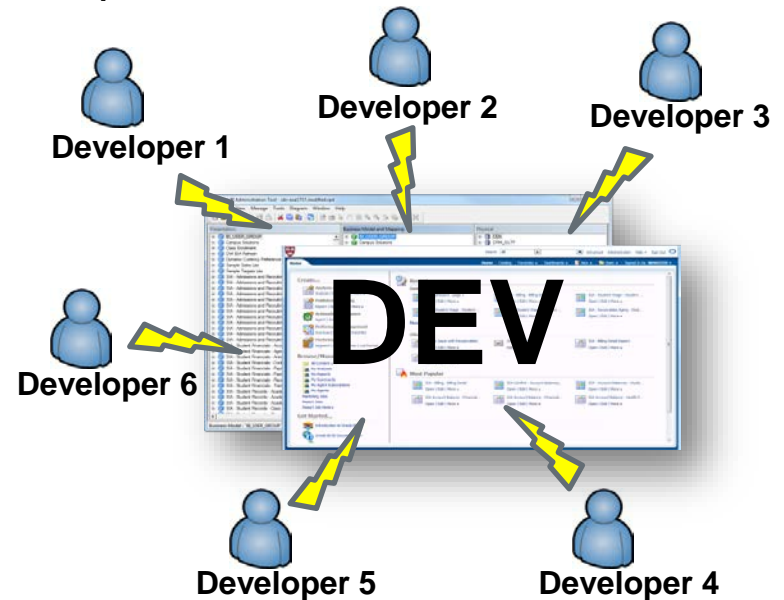
Where We Were

- 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



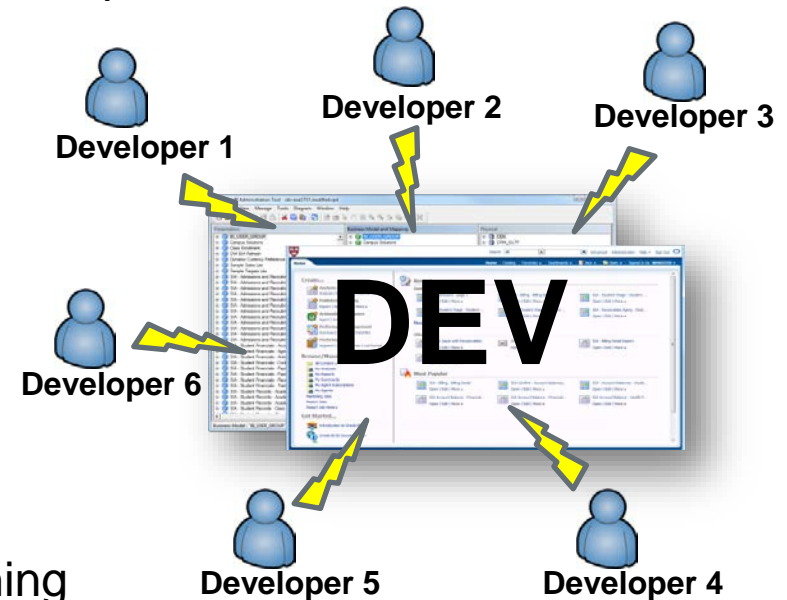
Where We Were

- 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



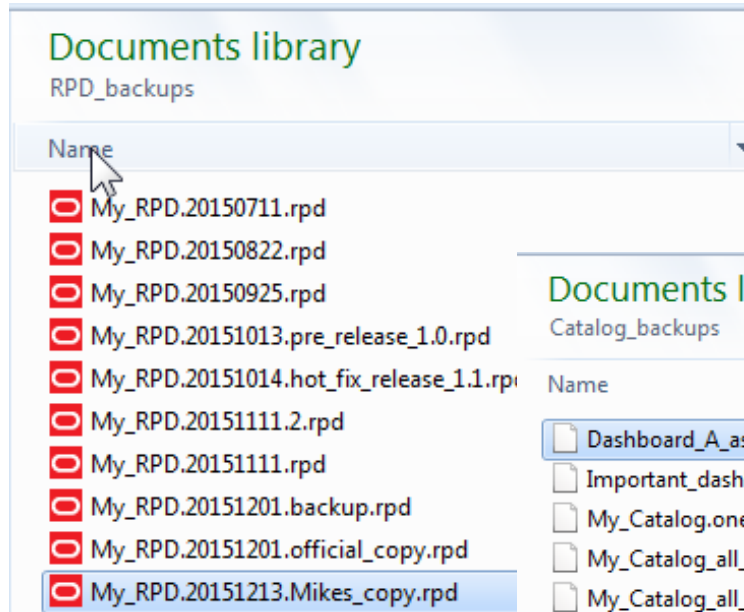
Where We Were

- 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

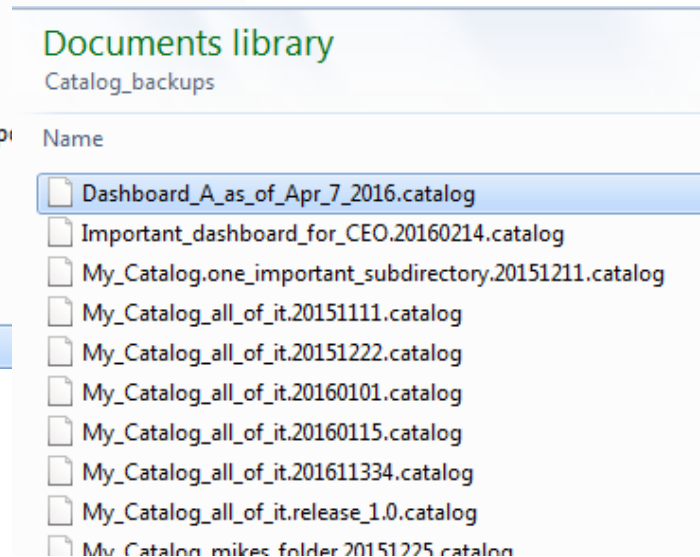


Where We Were

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

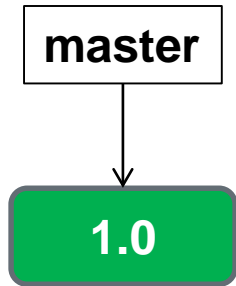
- “Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.”*
- Works well for text files; not so well for binaries

About Git

- Git is a version control system created by Linus Torvalds to maintain the Linux kernel
- It has many fans
- Why Git? imgtfy.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

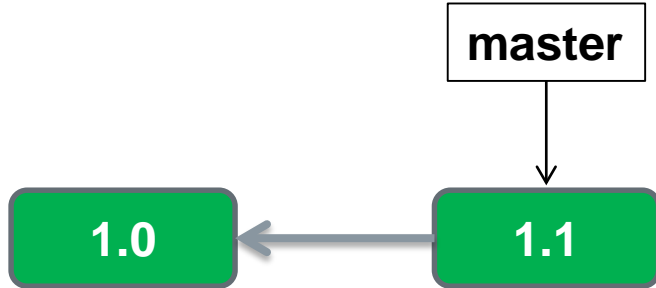
How We Got There: Overview

Branching and Merging in a Nutshell



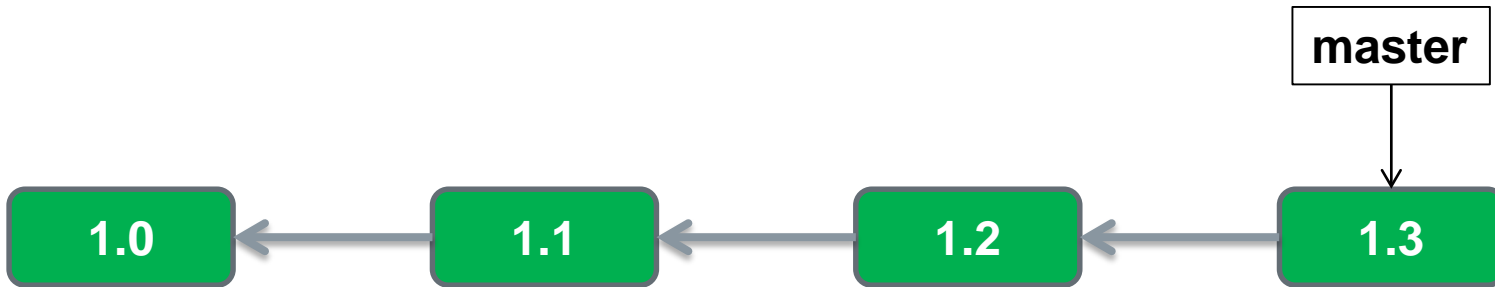
How We Got There: Overview

Branching and Merging in a Nutshell



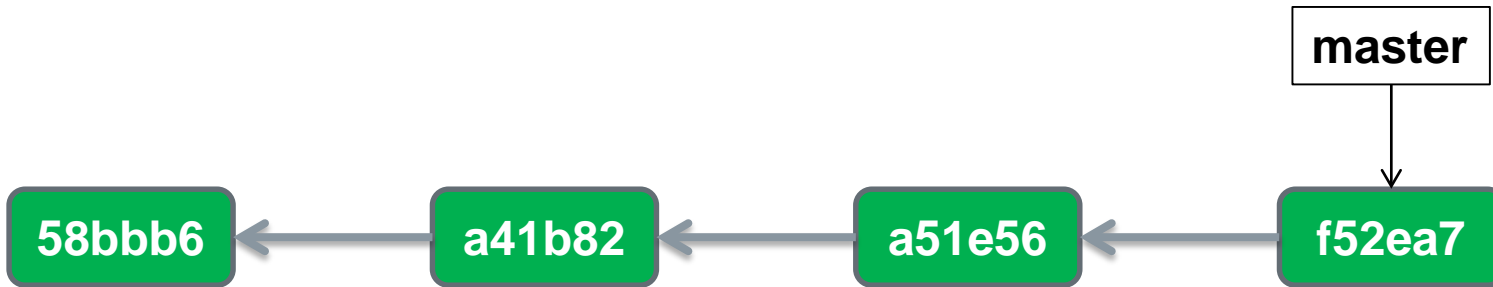
How We Got There: Overview

Branching and Merging in a Nutshell



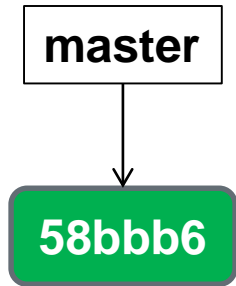
How We Got There: Overview

Branching and Merging in a Nutshell



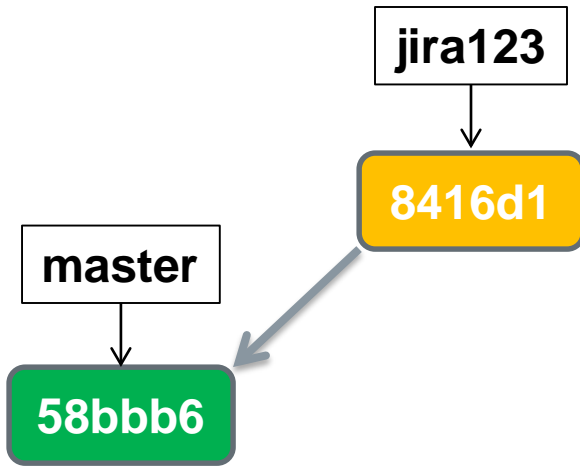
How We Got There: Overview

Branching and Merging in a Nutshell



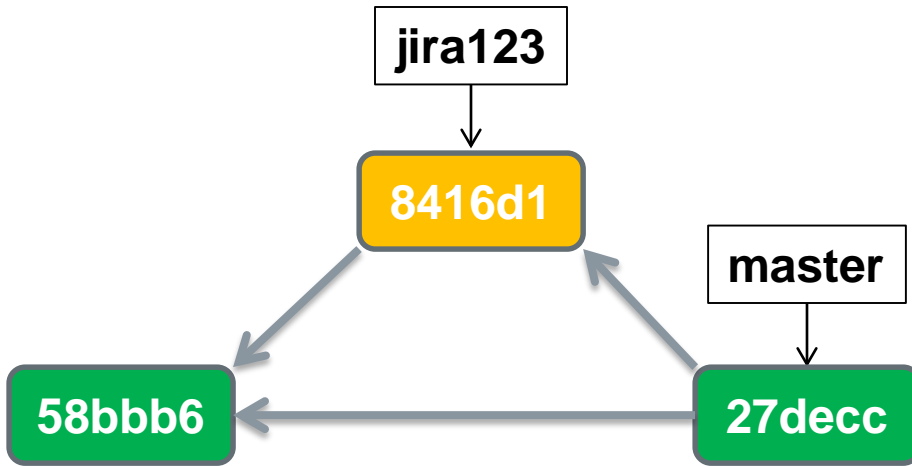
How We Got There: Overview

Branching and Merging in a Nutshell



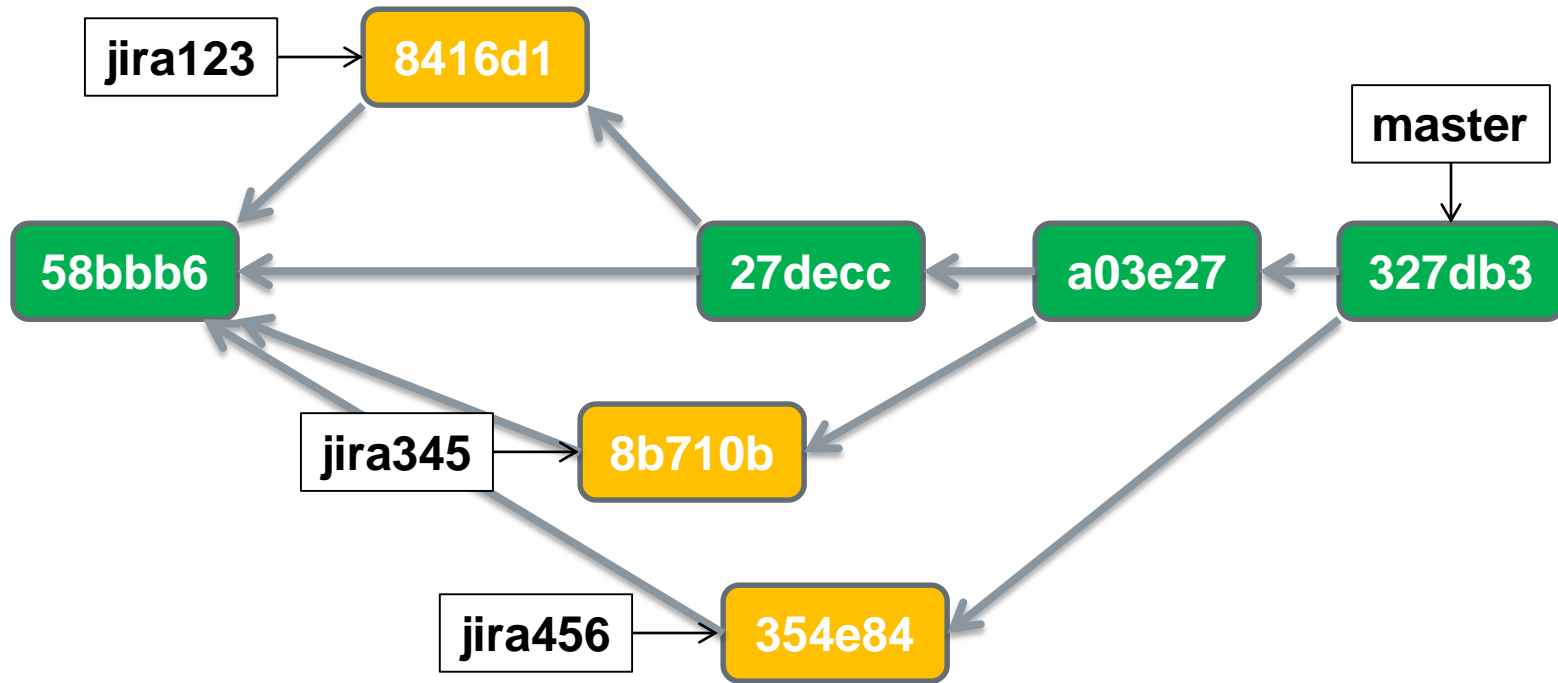
How We Got There: Overview

Branching and Merging in a Nutshell



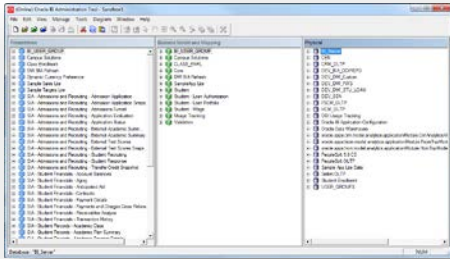
How We Got There: Overview

Branching and Merging in a Nutshell



How We Got There: Overview

Ok so how do we apply this to OBIEE?



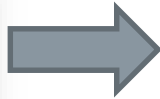
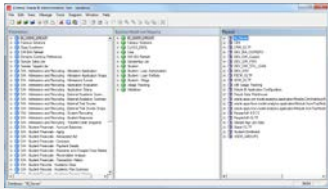
git

How We Got There: Overview

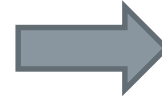
- Fortunately, the catalog consists primarily of text files so we can essentially just go ahead and put the catalog, or at least parts of the catalog, in Git



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



```
51 <!DOCTYPE rpd SYSTEM "http://www.ibm.com/xml/schemas/ibm-00000000.rpd" >
52 <?xml version="1.0" encoding="UTF-8" ?>
53 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
54 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
55 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
56 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
57 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
58 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
59 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
60 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
61 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
62 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
63 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
64 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
65 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
66 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
67 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
68 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
69 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
70 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
71 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
72 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
73 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
74 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
75 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
76 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
77 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
78 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
79 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
80 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
81 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
82 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
83 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
84 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
85 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
86 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
87 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
88 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
89 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
90 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
91 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
92 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
93 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
94 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
95 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
96 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
97 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
98 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
99 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
100 <EdgeTextDescr id="00000000-0000-0000-0000-00000000" name="Edge" />
```



How We Got There: Overview

Now that we've got that figured out...

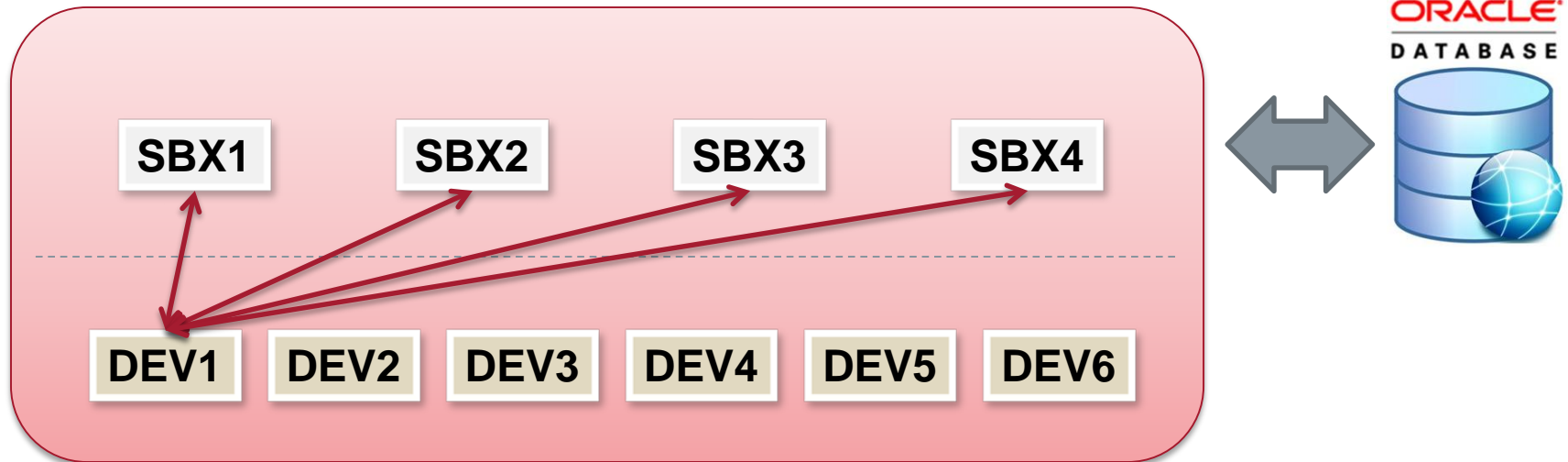
Lets talk Sandboxes

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

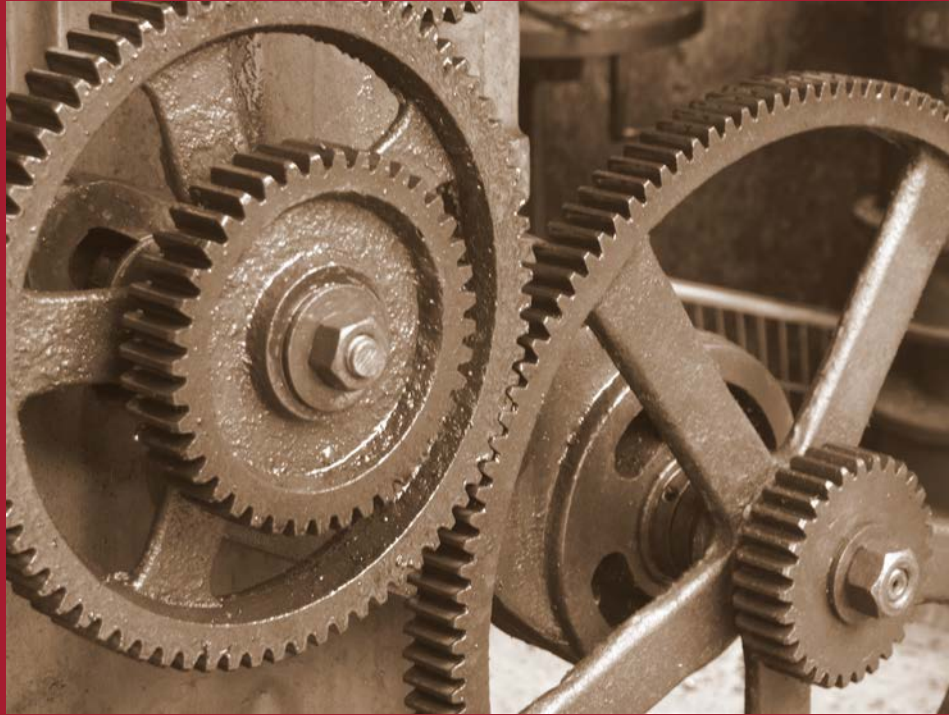
How We Got There: Overview

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

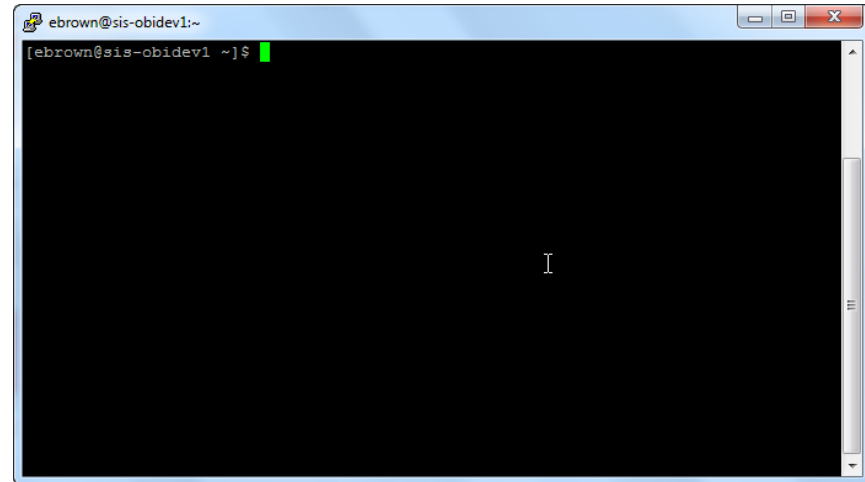
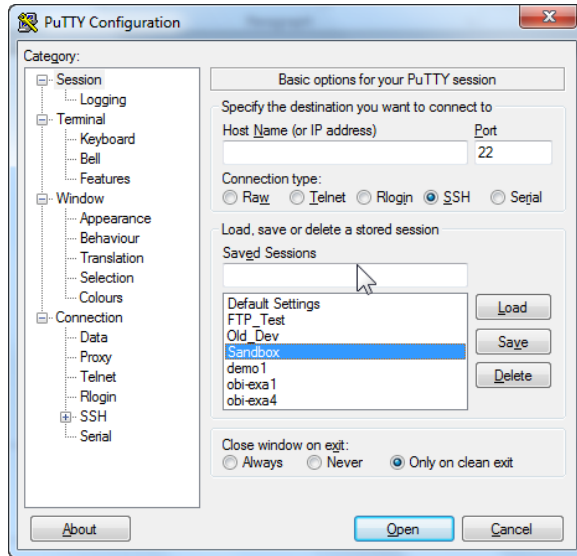
5. Pre-commit

6. Commit

The Development Process in Action

Development Process Steps

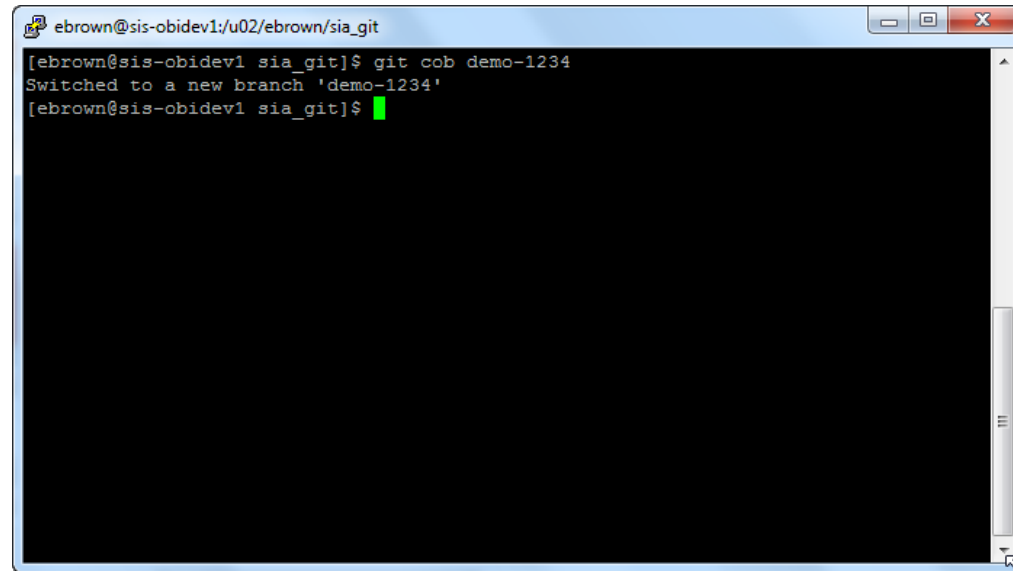
1. Log in to server



The Development Process in Action

Development Process Steps

2. Checkout or create branch

A terminal window with a blue title bar containing the text 'ebrown@sis-obidev1:/u02/ebrown/sia_git'. The terminal output shows the command 'git cob demo-1234' being executed, followed by the message 'Switched to a new branch 'demo-1234'' and a green cursor on the next line.

```
ebrown@sis-obidev1:/u02/ebrown/sia_git
[ebrown@sis-obidev1 sia_git]$ git cob demo-1234
Switched to a new branch 'demo-1234'
[ebrown@sis-obidev1 sia_git]$ █
```

The Development Process in Action

Development Process Steps

3. Deploy



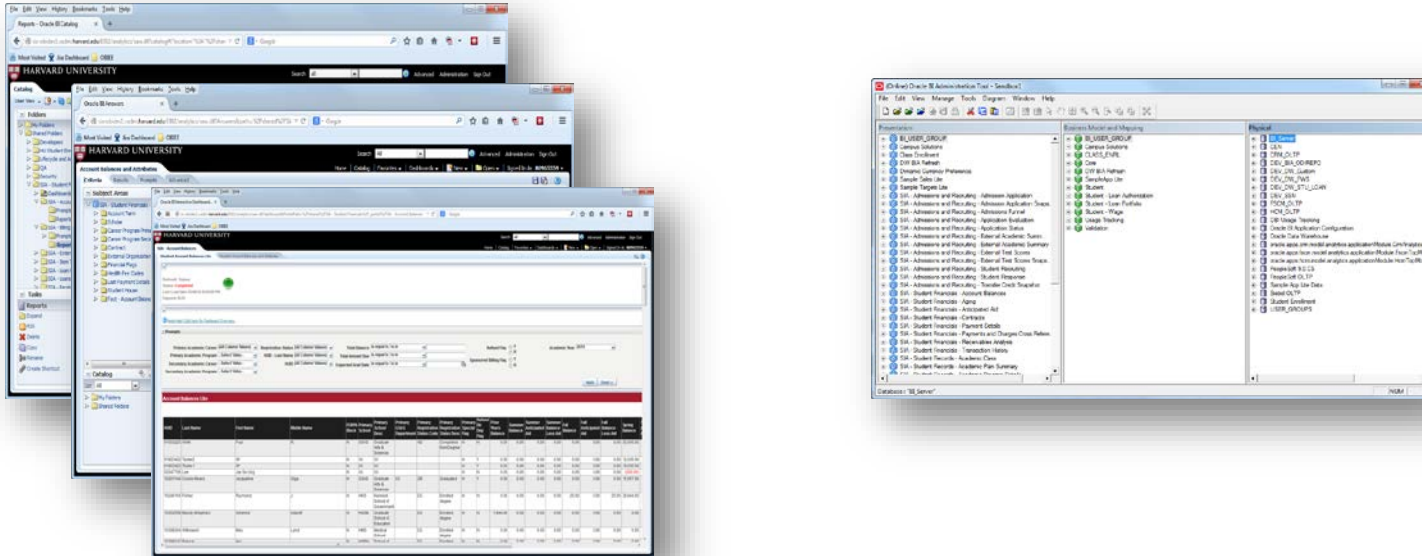
```
ebrown@sis-obidev1:/u02/ebrown/sia_git
[ebrown@sis-obidev1 sia_git]$ git cob demo-1234
Switched to a new branch 'demo-1234'
[ebrown@sis-obidev1 sia_git]$ deploy
```

A terminal window with a blue title bar containing the text 'ebrown@sis-obidev1:/u02/ebrown/sia_git'. The terminal content shows the execution of 'git cob demo-1234', which results in 'Switched to a new branch 'demo-1234'', followed by the 'deploy' command. A green cursor is visible at the end of the 'deploy' line. The window has standard Linux window controls (minimize, maximize, close) in the top right corner and a vertical scrollbar on the right side.

The Development Process in Action

Development Process Steps

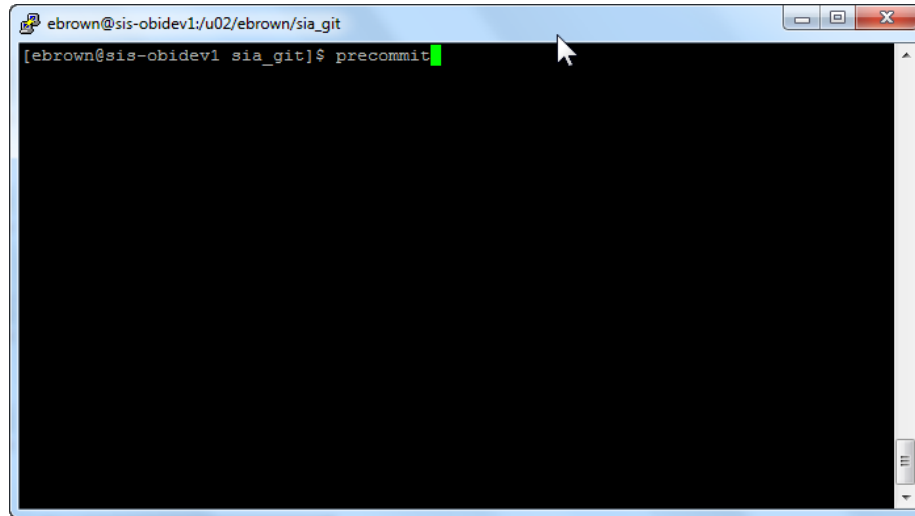
4. Develop



The Development Process in Action

Development Process Steps

5. Pre-commit



```
ebrown@sis-obidev1:/u02/ebrown/sia_git
[ebrown@sis-obidev1 sia_git]$ precommit
```

A terminal window with a blue title bar. The title bar text is "ebrown@sis-obidev1:/u02/ebrown/sia_git". The terminal content shows a prompt "[ebrown@sis-obidev1 sia_git]" followed by the command "precommit" and a green cursor. The window has standard Linux window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

The Development Process in Action

Development Process Steps

6. Commit



```
ebrown@sis-obidev1:/u02/ebrown/sia_git
[ebrown@sis-obidev1 sia_git]$ git commit
```

A terminal window with a blue title bar. The title bar text is "ebrown@sis-obidev1:/u02/ebrown/sia_git". The terminal content shows the command "[ebrown@sis-obidev1 sia_git]\$ git commit" with a green cursor at the end. The rest of the terminal is black with a white cursor "I" at the bottom center.

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

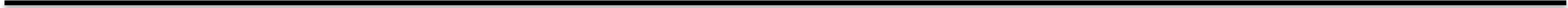
1. Log in to server

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

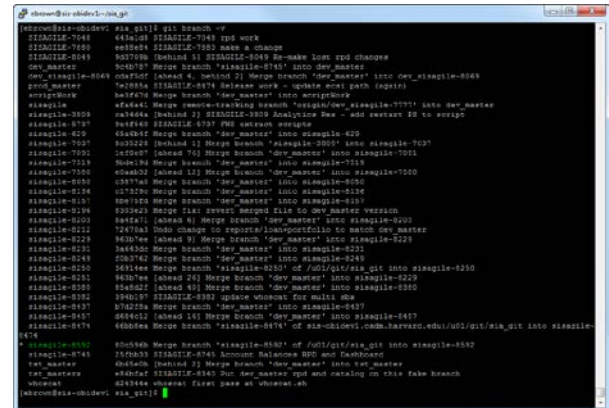
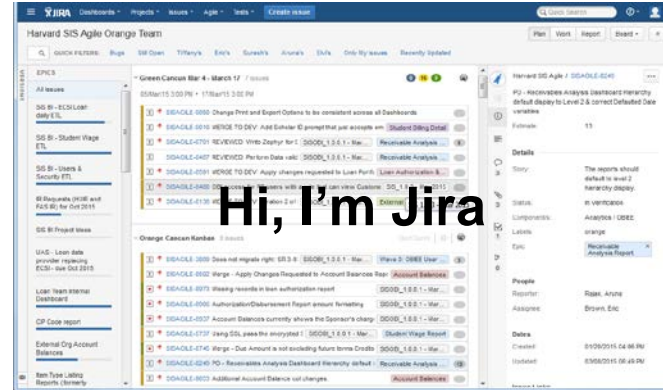
1. Develop
2. Peer Review
3. Merge to master
4. PO/Manager Review
5. Release to Test
- 6. Release to Prod**

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



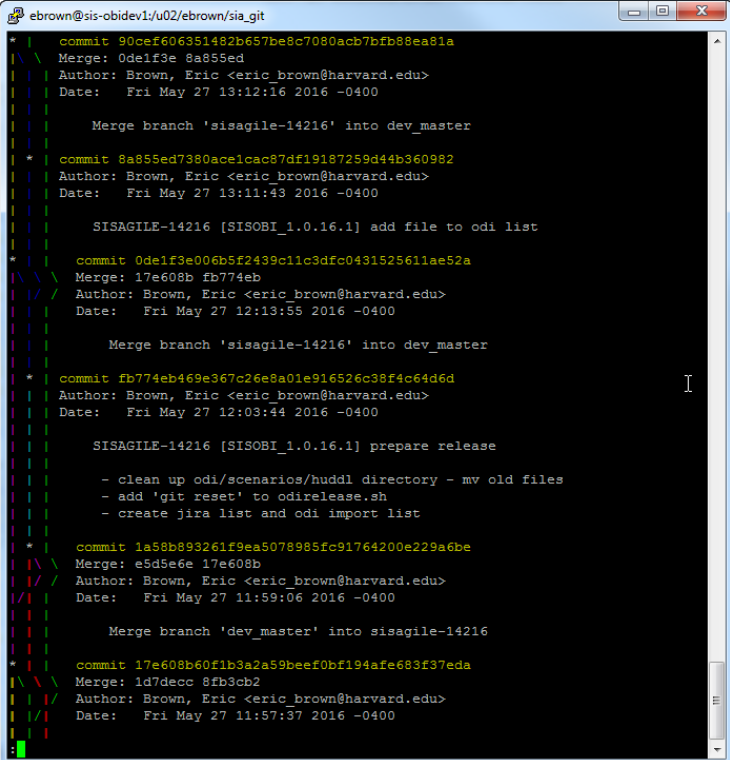
The Development Process in Action

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.



```
ebrown@sis-obidev1:/u02/ebrown/sia_git
* commit 90cef606351482b657be8c7080acb7bfb88ea81a
  Merge: 0def13e 8a855ed
  Author: Brown, Eric <eric_brown@harvard.edu>
  Date: Fri May 27 13:12:16 2016 -0400

    Merge branch 'sisagile-14216' into dev_master
* commit 8a855ed7380ace1cac87df19187259d44b360982
  Author: Brown, Eric <eric_brown@harvard.edu>
  Date: Fri May 27 13:11:43 2016 -0400

    SISAGILE-14216 [SISOBI_1.0.16.1] add file to odi list
* commit 0def13e006b5f2439c11c3dfc0431525611ae52a
  Merge: 17e608b fb774eb
  Author: Brown, Eric <eric_brown@harvard.edu>
  Date: Fri May 27 12:13:55 2016 -0400

    Merge branch 'sisagile-14216' into dev_master
* commit fb774eb469e367c26e9a01e916526c38f4c64d6d
  Author: Brown, Eric <eric_brown@harvard.edu>
  Date: Fri May 27 12:03:44 2016 -0400

    SISAGILE-14216 [SISOBI_1.0.16.1] prepare release
    - clean up odi/scenarios/huddl directory - mv old files
    - add 'git reset' to odirelease.sh
    - create jara list and odi import list
* commit 1a58b893261f9ea5078985f91764200e229a6be
  Merge: e5d5e6e 17e608b
  Author: Brown, Eric <eric_brown@harvard.edu>
  Date: Fri May 27 11:59:06 2016 -0400

    Merge branch 'dev_master' into sisagile-14216
* commit 17e608b60f1b3a2a59bee0bf194afe683f37eda
  Merge: 1d7decc 8fb3cb2
  Author: Brown, Eric <eric_brown@harvard.edu>
  Date: Fri May 27 11:57:37 2016 -0400
```

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

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