

Project Management

1

Overview

- How to manage a project?
- What is software configuration management?
- Version control systems
- Issue tracking systems

N. Meng, L. Zhang

2

2

What is Project Management?

- Effective project management focuses on the 4 P's:
 - People: the most important element
 - recruiting, training, performance management
 - Product: the software to build
 - Project objectives, scope, alternative solutions
 - Process: define activities and tasks involved
 - Milestones, work products, QA points
 - Project: progress control
 - Planning, monitoring, controlling

N. Meng, L. Zhang

3

3

The "First Law"

- No matter where you are in the system life cycle, **the system will change, and the desire to change it will persist throughout the life cycle.**

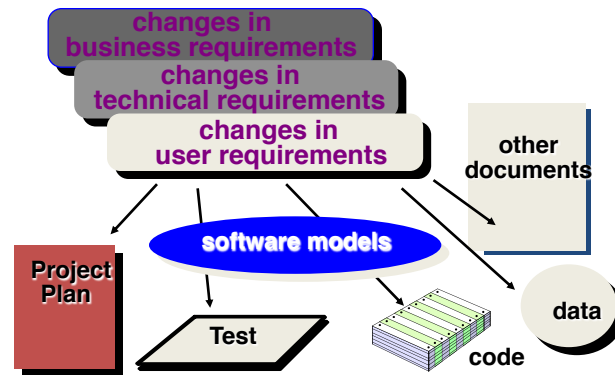
Bersoff, et al, 1980

N. Meng, L. Zhang

4

4

What Are These Changes?



N. Meng, L. Zhang

5

5

Software Configuration Management (SCM)

- Definition
 - The task of tracking and controlling changes in software
- SCM repository
 - tools that allow developers to effectively manage changes
 - Version control system
 - Issue tracking system

N. Meng, L. Zhang

6

6

Version Control System

7

What Is Version Control System?

- VCS, also known as Revision Control System
- To manage changes to documents, programs, large websites, and other collections of information
 - CVS, SVN, Mercurial, GIT

N. Meng, L. Zhang

8

8

What Do We Mean by "Manage Changes" ?

- What changes have been made?
- Why are the changes made?
- Who makes the changes?
- Can we redo/undo some changes?
- Can we branch the project?



N. Meng, L. Zhang

9

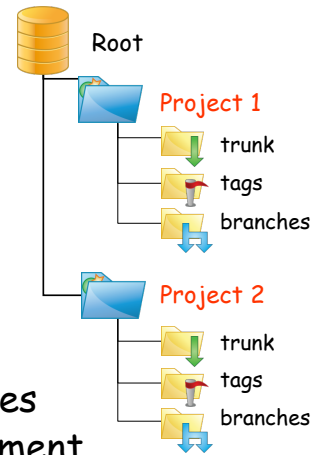
9

Subversion Version Control System (SVN)

10

Subversion Repository Layout

- One SVN server can hold many repositories
- One repository can hold many projects
- One project contains
 - Trunk: Main line of development
 - Tags: Markers to highlight notable revisions—major releases
 - Branches: Side lines of development



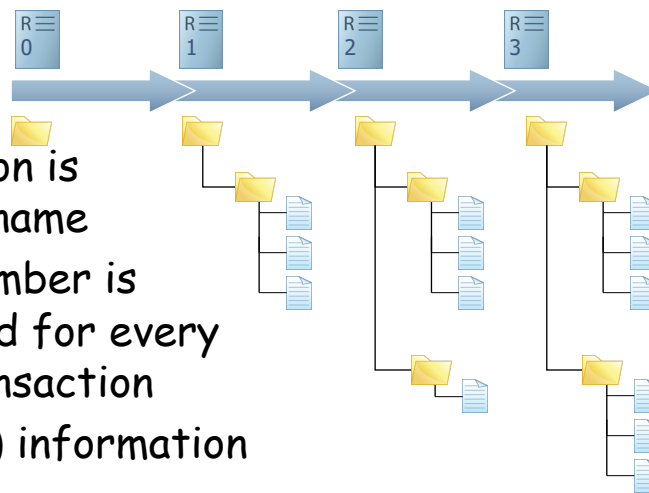
N. Meng, L. Zhang

11

11

Each project has multiple revisions

- Each revision is assigned a name
- Revision number is incremented for every commit transaction
- Delta (diff) information is recorded



N. Meng, L. Zhang

12

12

Basic Features of a Repository

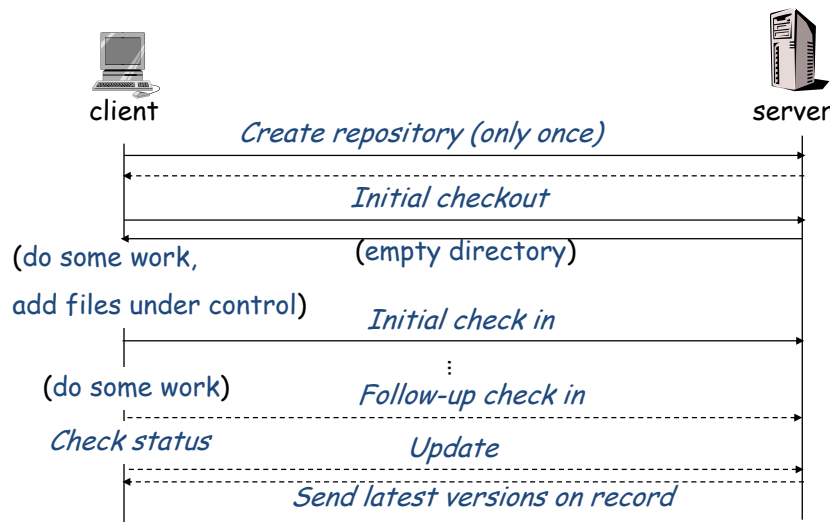
- Keep the history of all changes to files and directories
 - You can add in new versions
 - You can recover any previous version
- Access control
 - Read/write permission for users
- Logging
 - Author, date, and reason for a change

N. Meng, L. Zhang

13

13

Typical Workflow



N. Meng, L. Zhang

14

14

Additional Features

- Diff
- Branch
- Merge

N. Meng, L. Zhang

15

15

Diff

- To display the differences between two revisions
 - **What has been changed?**
 - Add or delete a line of text
 - No update, or move

```
Version 1:          Version 2:
x = 0;              x = 1;
y = 1;              y = 1;
Diff:
- x = 0;
+ x = 1;
```

N. Meng, L. Zhang

16

16

Key Points about Diff

- A key operation of version control systems
- A lot of features are based on diff
 - Save new versions
 - Recover a prior version
 - Patch
- We use $\text{Diff}(v1, v2)$ to represent changes on $v1$ for $v2$
 - $\text{Diff}(v1, v2) \neq \text{Diff}(v2, v1)$

N. Meng, L. Zhang

17

17

Diff: a Real Example

```

Index: trunk/compiler/org/eclipse/jdt/internal/compiler/ast/Expression.java
-----
--- trunk/compiler/org/eclipse/jdt/internal/compiler/ast/Expression.java      (revision 9042)
+++ trunk/compiler/org/eclipse/jdt/internal/compiler/ast/Expression.java      (revision 9043)
@@ -223,7 +223,7 @@
     this.implicitConversion = (runtimeTimeType.id << 4) + compileTimeType.id;
     break;
     default : // regular object ref
     if (compileTimeType.isRawType() && runtimeTimeType.isParameterizedType())
     if (compileTimeType.isRawType() && runtimeTimeType.isBoundParameterizedType())
     scope.problemReporter().unsafeRawExpression(this, compileTimeType);
     }
-----

```

Start line in the old version Start line in the new version

- `svn diff -r v1:v2 filename`
- "+": added lines, "-": deleted lines
- Some unchanged lines are shown to indicate program context

N. Meng, L. Zhang

18

18

Changes Detected by Diff

- Addition/Deletion of directories
- Addition/Deletion of files
- A renamed file is reported as a separate addition and a separate deletion
- Addition/Deletion of lines

N. Meng, L. Zhang

19

19

Branch

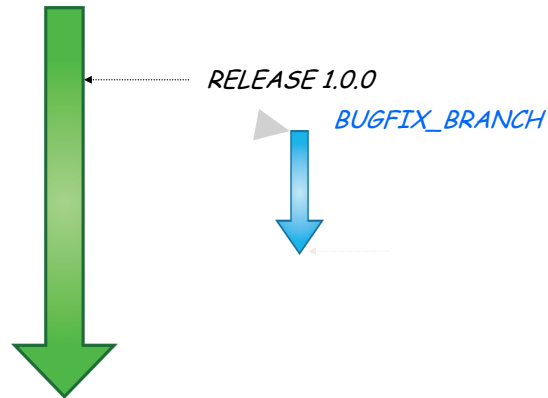
- Scenario
 - You deliver a great product to your customers: REL-1.0.0
 - Your development team continue adding new features on the **trunk**
 - Customers report a major bug in the product and ask for a fix
 - What do you do?

N. Meng, L. Zhang

20

20

Branch and patch separately!



- svn copy path/to/trunk path/to/branch

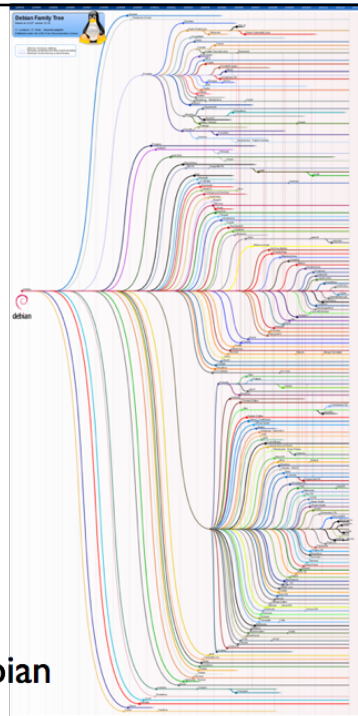
N. Meng, L. Zhang

21

21

Other reasons to branch

- Separate branches for
 - Tentative new features
 - Different products
 - Different teams
 - Different releases
- Where to put the major development, branch, trunk, both?



N. Meng, L. Zhang

Debian

22

Pros and Cons of Branch

- Pros
 - Separation of concerns among teams and developers
 - Parallel version history without interference between branches
- Cons
 - Branches may diverge a lot
 - Hard to propagate changes across branches

N. Meng, L. Zhang

23

23

Merge

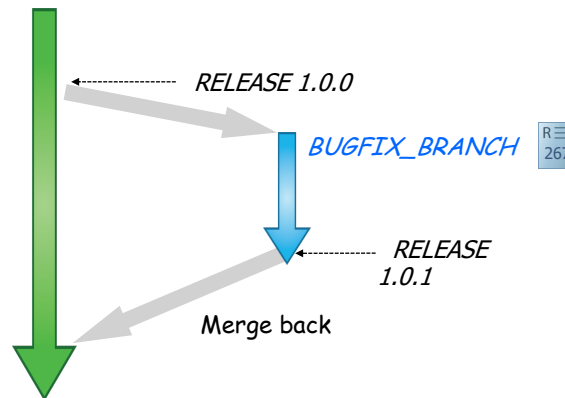
- Scenario
 - After fixing the major bug on a branch, you have to apply the same/similar changes to the trunk
 - What do you do?

N. Meng, L. Zhang

24

24

Merge back the patch!



- `svn merge -reintegrate path/to/branch`

N. Meng, L. Zhang

25

25

What can happen when merging?

- Conflict
 - Two people edit the same file

```
void f(int i) {
<<<<<<<< .mine
int j = 3;
=====
int j = 4;
>>>>>>> .r13
```

- Resolve the conflict manually and checked in again

N. Meng, L. Zhang

26

26

Distributed Version Control: GIT

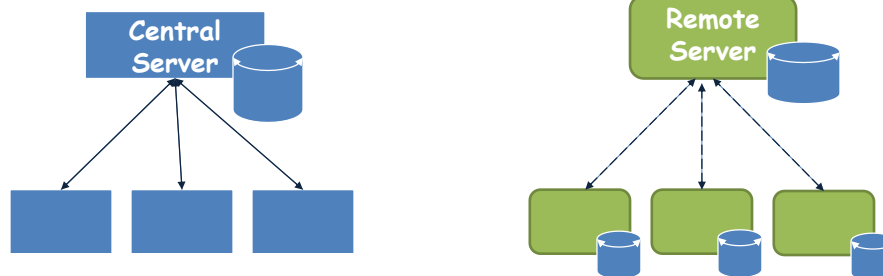
- Everyone has their own local version control repository
 - Like a local branch of the project
 - Remote updates and commits are like branch merge
 - Local commits used to backup projects
 - Github allows developers to contribute by working on branches

N. Meng, L. Zhang

27

27

Centralized VC vs. Distributed VC[1]



N. Meng, L. Zhang

28

28

Git Initialization [1]

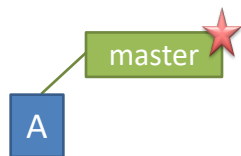
```
C:\> mkdir CoolProject
C:\> cd CoolProject
C:\CoolProject > git init
Initialized empty Git repository in
C:/CoolProject/.git
C:\CoolProject > notepad README.txt
C:\CoolProject > git add .
C:\CoolProject > git commit -m 'my first commit'
[master (root-commit) 7106a52] my first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.txt
C:\CoolProject > git remote add origin remote
repository URL
# Sets the new remote
C:\CoolProject > git push origin master
# Pushes the changes in your local repository to
the remote repository
```

N. Meng, L. Zhang

29

29

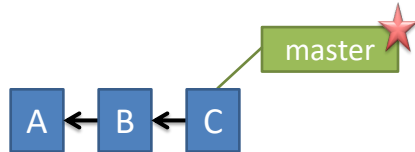
Git Branch & Merge [1]



```
> git commit -m 'my first commit'
```

30

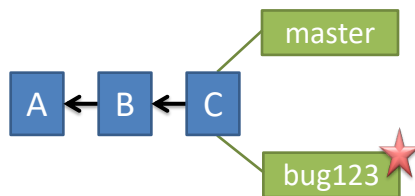
Branches Illustrated [1]



```
> git commit (x2)
```

31

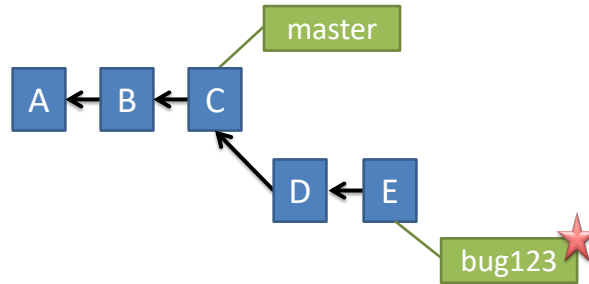
Branches Illustrated [1]



```
> git checkout -b bug123
```

32

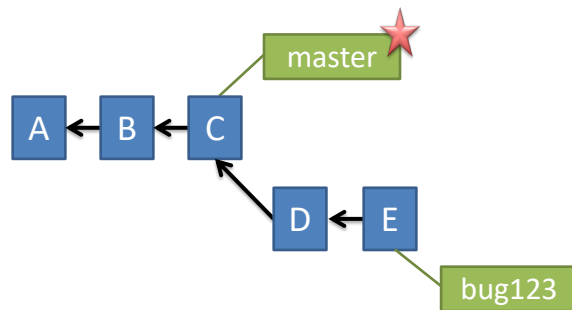
Branches Illustrated



```
> git commit (x2)
```

33

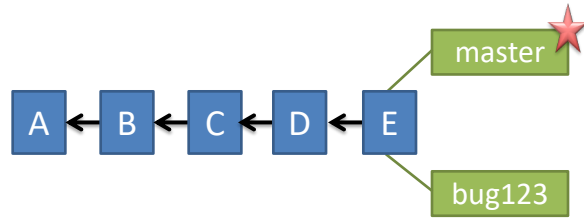
Branches Illustrated



```
> git checkout master
```

34

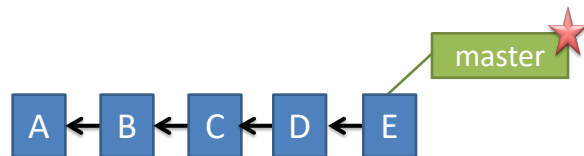
Branches Illustrated



```
> git merge bug123
```

35

Branches Illustrated



36

Tips for Version Control

- **Small commits**
 - Check in logically relevant changes as a commit
- **Write meaningful commit messages**
 - Facilitate change understanding, applying, and reverting
- **Avoid commit noise**
 - Commit compilable or even deliverable code

N. Meng, L. Zhang

37

37

Issue Tracking System

38

What Is Issue Tracking System?

- ITS, also known as trouble ticket system, support ticket, request management, or incident ticket system
- Manages and maintains lists of issues, as needed by an organization
 - To create, update, and resolve reported issues by customers or developers
 - Bugzilla, JIRA

N. Meng, L. Zhang

39

39

What Do We Mean by "Issues"?

- A unit of work to accomplish an improvement in a system
- It could be
 - a bug
 - a requested feature
 - a patch
 - missing documentation, ...

N. Meng, L. Zhang

40

40

Why Do We Need Issue Tracking?

- Developers need communication while making changes
 - Mailing List
 - Hard to manage, come with all other mails
 - Not well organized
 - Forum
 - Categorized by topic
 - Notify people when a reply is posted
 - No track to code and issue status

N. Meng, L. Zhang

41

41

What Is Included in An Issue?

Agile Board
Export

Details

Type: Documentation

Status: OPEN

Priority: Minor

Resolution: Unresolved

Affects Version/s: 0.11.0

Fix Version/s: None

Component/s: Mahout spark shell

Labels: None

People

Assignee: Unassigned

Reporter: Sergey Tryuber

Votes: 0 [Vote for this issue](#)

Watchers: 2 [Start watching this issue](#)

Description

There is a bug in [documentation](#) (2.3.5 Collecting to HDFS).
Instead of:

Dates

Created:

N. Meng, L. Zhang

42

42

Basic Features

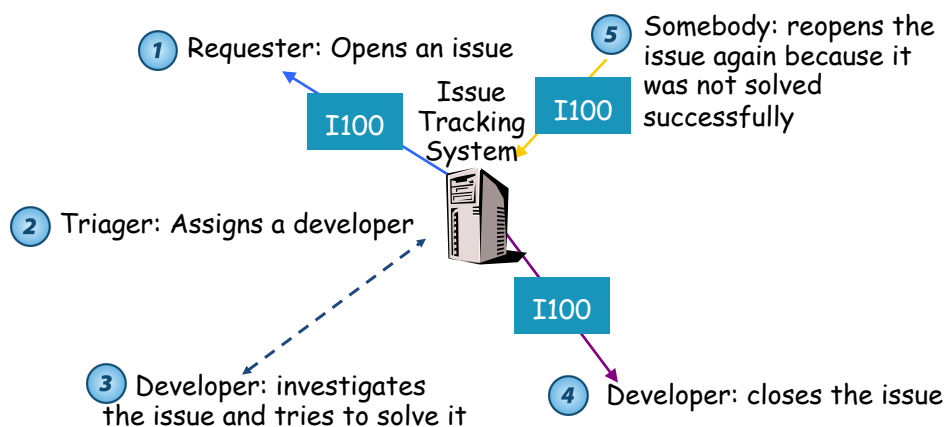
- Structurally describe issues
 - Solving status, severity levels
- Track status of the issue
- Assign a unique ID to each issue
 - Some system automates connection between commit and issue via issue ID

N. Meng, L. Zhang

43

43

Typical Workflow



N. Meng, L. Zhang

44

44

Resolution of An Issue

- Fixed
 - A bug is fixed, a feature is added, a patch is applied
- Invalid
 - Bug cannot be reproduced, features do not make sense, patch is not correct
- Duplicate
 - It is a duplicate of an existing issue
 - Get merged with the other issue

N. Meng, L. Zhang

45

45

Resolution of An Issue

- Won't fix
 - The developers decide not to fix the bug or accommodate the new feature
 - Limited human resource, lack of essential information to reproduce a bug, lack of expertise

N. Meng, L. Zhang

46

46

Issue Tracking & Version Control

- Many project hosting websites include issue tracking systems
 - Google Code
 - Github
 - BitBucket
 - Sourceforge

N. Meng, L. Zhang

47

47

Reference

[1] Mark Groves, Introducing Git version control into your system, PPT

N. Meng, L. Zhang

48

48