Ohjelmankehityspr., versionhallinta ja testaus

Introduction to Version Control – Chapter 1

KAJAANIN
AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Deepak K.C. ; deepak.kc@kamk.fi ;

Version Control

Software has multiple versions

- Different releases of a product
- Variations for different platforms
- Version with a development cycle
  - Test release with debugging mode
  - Alpha, beta of final release

Version control

- A system that records changes to a file or set of files over time
- It tracks multiple versions
- Possible to recall specific versions later
  - To revert files to a previous state
  - Multiple versions to exist simultaneously
  - Compare changes over time
  - Modification history

Why Version Control?

- Because everyone use it
- A basic software development tool
- A place to store your code
- Records of software evolution
- Synchronization between developers
- Facilitates automated build, test, deploy
- Developer not tied to one machine

Basic concepts

- Tracking changes
- Committing
- Revisions and changesets
- Gettting updates
- Conflicts
- Diffing (viewing the differences)
- Branching & merging

Tracking changes

- Tracking changes within directories or files
- Knowing a file changed to knowing specific characters or bytes in a file have changed
- You need to specify a directory or set of files that should have their changes traacked by VC
  - It happens by **checking out** (or cloning) a repository from a host or
  - Telling the software which of your files you wish to have under version control
- Set of files or directories under version control are commonly called a **repository**

Committing

- Each change is tracked automatically
- Changes can be
  - Modifying a file, deleting a directory, adding a new file, moving files or just anything that matters the sate of the file
- Instead of recording each change individually, the version control
  - Will wait for you to submit your changes as a single collection of actions
  - This collection of actions is known as commit

Revisions & changesets

- When a commit is made
  - Changes are reorded as a changset & given a unique revision
  - Revision can be
    - In the form of incremented number (1,2,3) or
    - Unique hash (846eee7d92415cfd3f8a936d9ba5c3ad345831e5)
    - Knowing the revision makes it easy to view or reference it later
  - Changeset includes
    - A reference to the person who made the commit
    - When the change was made
    - The files or directories affected
    - Comment and changes with the files

Getting updates

- Having the latest version reduces the chance of conflict
- Requesting update or pull will download only the changes since your last request

Diffing

- Viewing the differences
- To view what changed between version
- To compare two files or even a set of files to see what lines of code changed, when it was changed and who change it

Conflicts

- When two users make similar kind of changes and the version control is not able to determine the correct & authoritative change, VC system provides a way to view the difference between the conflicting versions and allows user to make a choice.
  - Possibility to edit files manually to merge the options or
  - Allowing one version to win over another

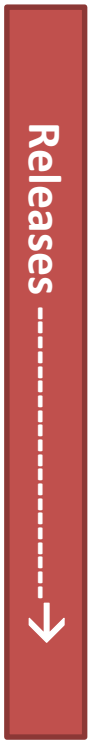
Types of version control systems

- Centralized version control
  - Works in a client server relationship
  - Repository located in one place and provides access to many clients
  - All changes, users, commits & information must be sent & received from the central repository
  - Example: subversion
- Distributed version control
  - Each user has their own copy on the entire repository
  - Network of individual repositories
  - Most popular distributed version control systems are git & mercurial

Scenario I - Development with bug

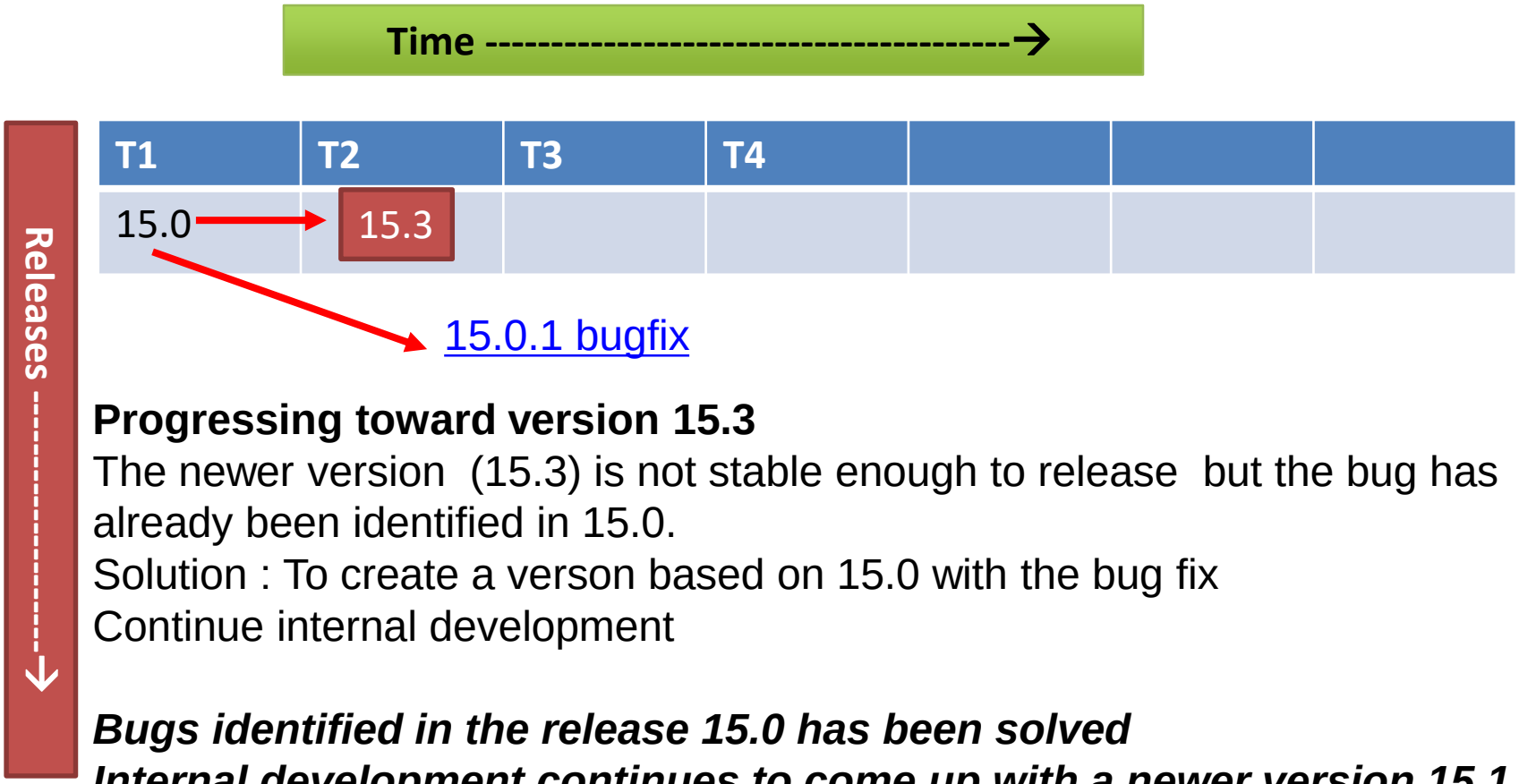
- First public release of a new application named ”application1”

T1	T2					
15.0						



15.0 is the first public release of application1
Bugs identified in the release
Internal development continues to come up with a newer version

Release with the bug fix indentified in version 1.0



Progressing toward version 15.3

The newer version (15.3) is not stable enough to release but the bug has already been identified in 15.0.

Solution : To create a verson based on 15.0 with the bug fix

Continue internal development

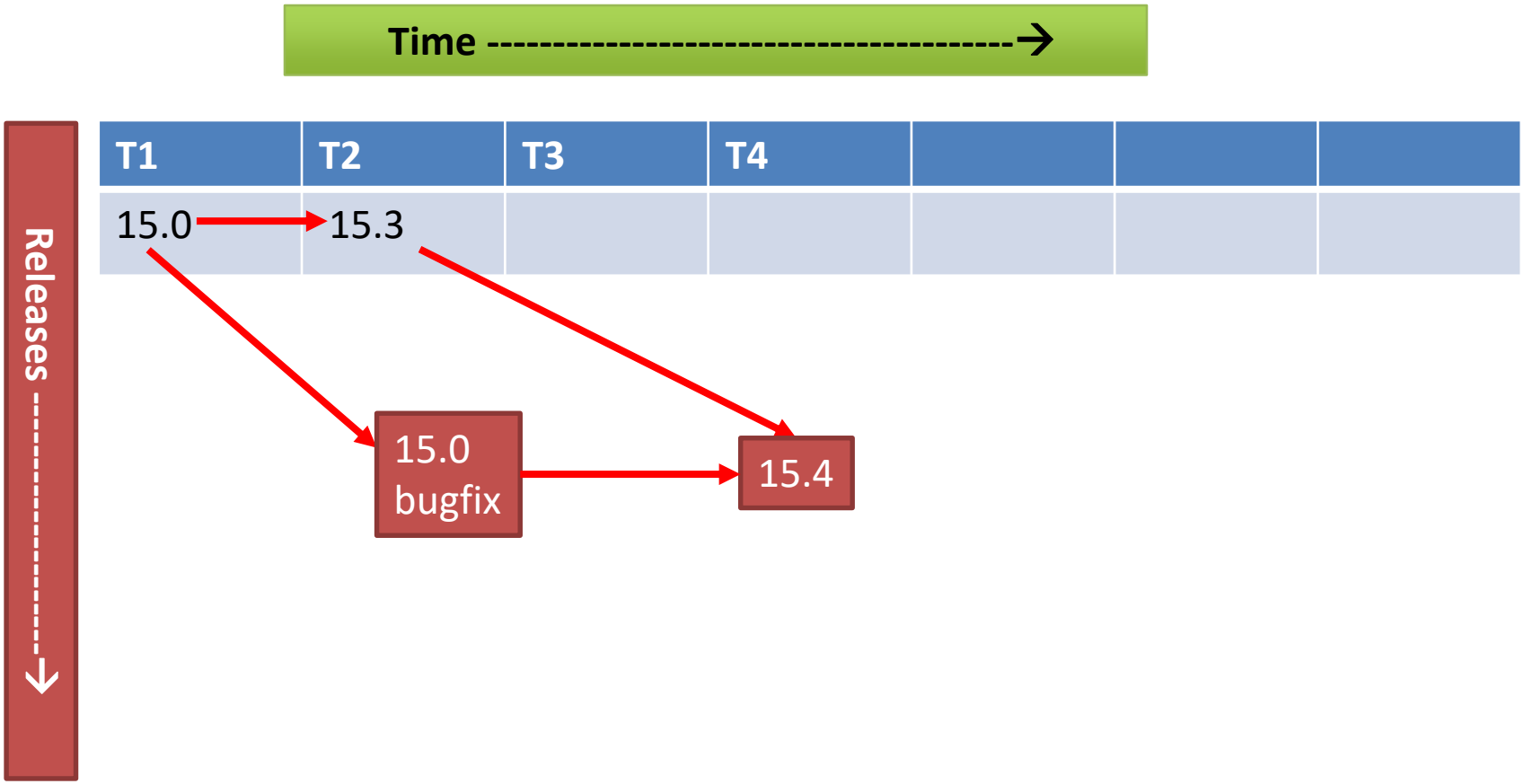
Bugs identified in the release 15.0 has been solved

Internal development continues to come up with a newer version 15.1

Note:

- there are now two lines of development beginning at 15.0. This is ***branching***.
- *The bug fix should also be applied to main code line so the next relase has the fix*

Applying bug fix to the main code line



Note:

- Two separate lines of development now come back together. This is **merging** or **updating**

References

- http://git-scm.com/doc
- http://guides.beanstalkapp.com/version-control/intro-to-version-control.html
- https://www.atlassian.com/git/tutorials/using-branches/