Introduction to CTIP

Lee, Dong-Ah

2014-03-12

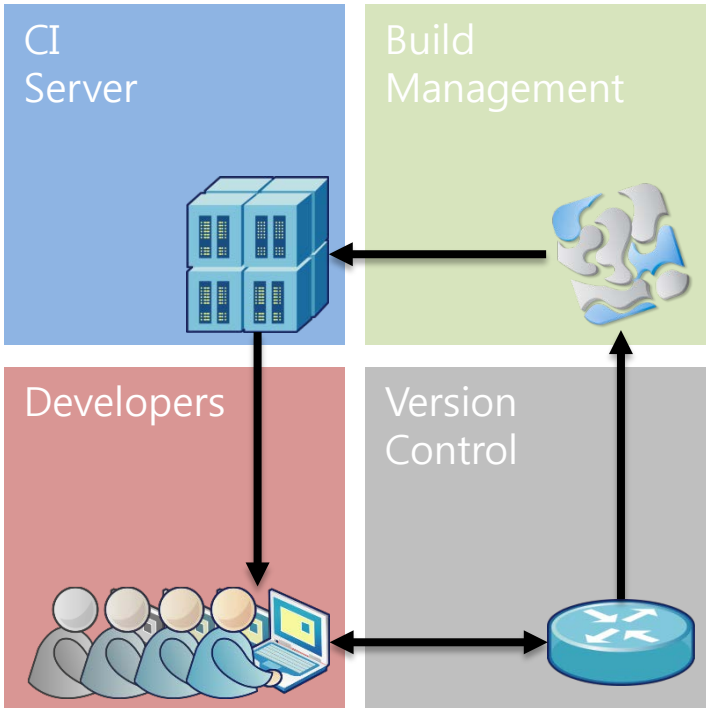
Contents

- CI?
- CTIP?
- COMPONENTS?
- ADVANTAGES!
- DISADVANTAGES?
- Team Projects

CI?

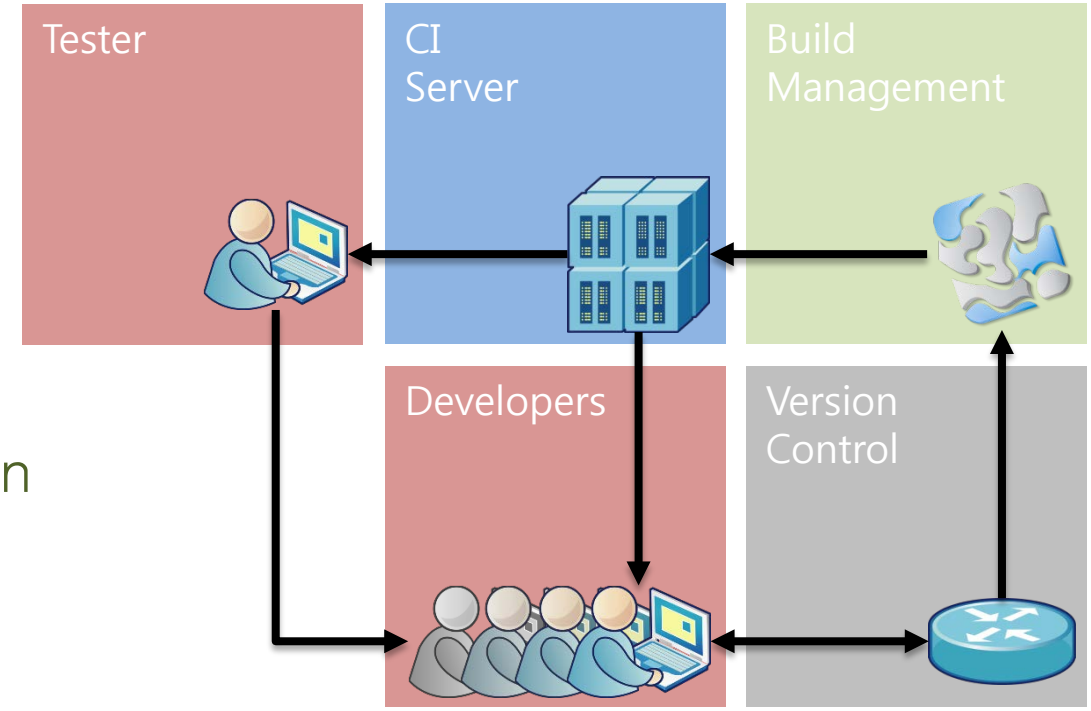
Continuous Integration

- As soon as the work on a task is complete, it **is integrated into the whole system.**
- After such integration, all the **unit tests** in the system must pass.



CTIP?

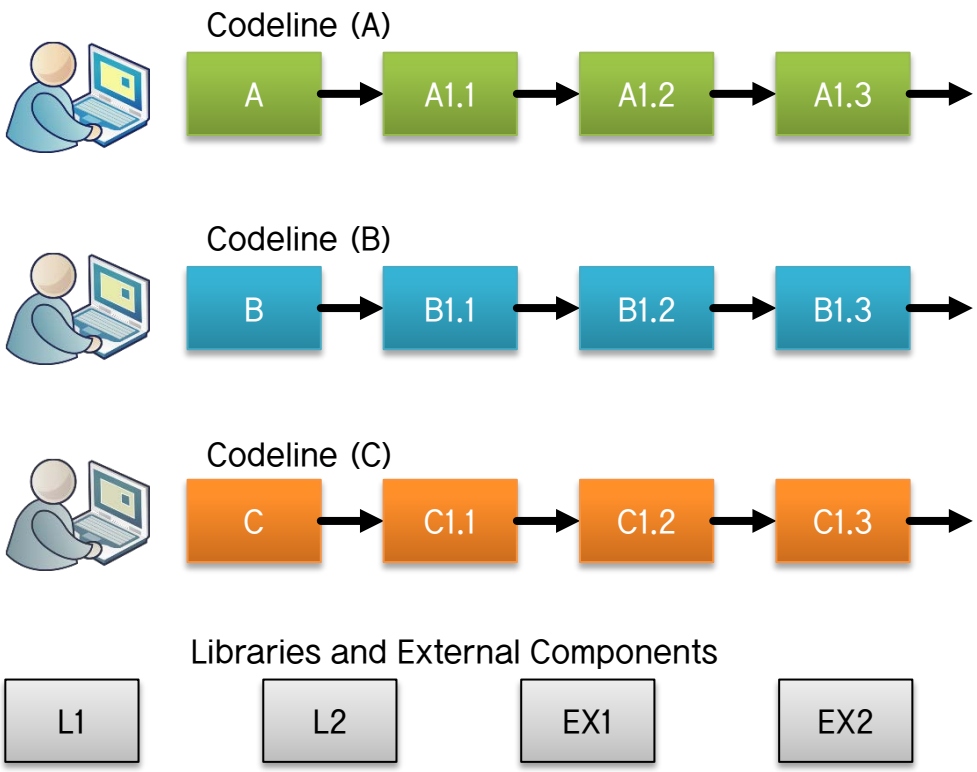
Continuous Test & Integration Platform

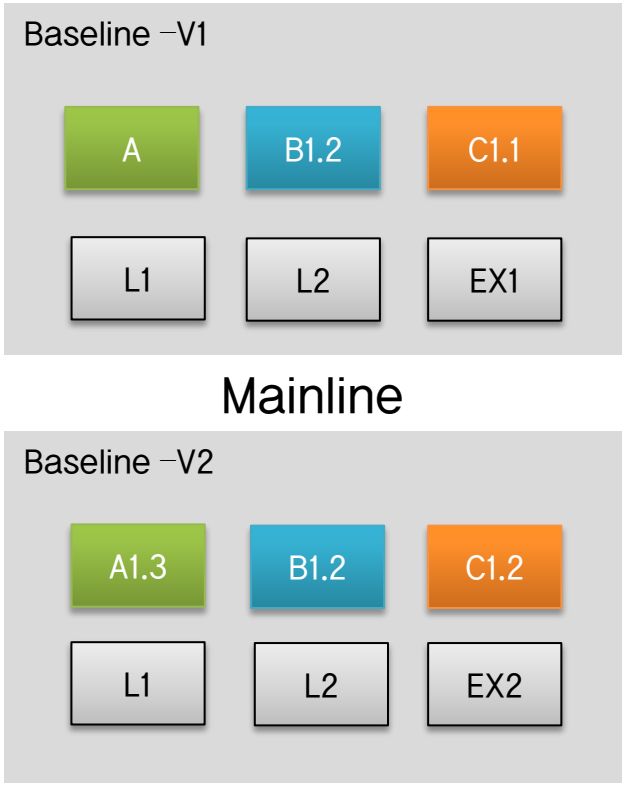


- Continuous Integration

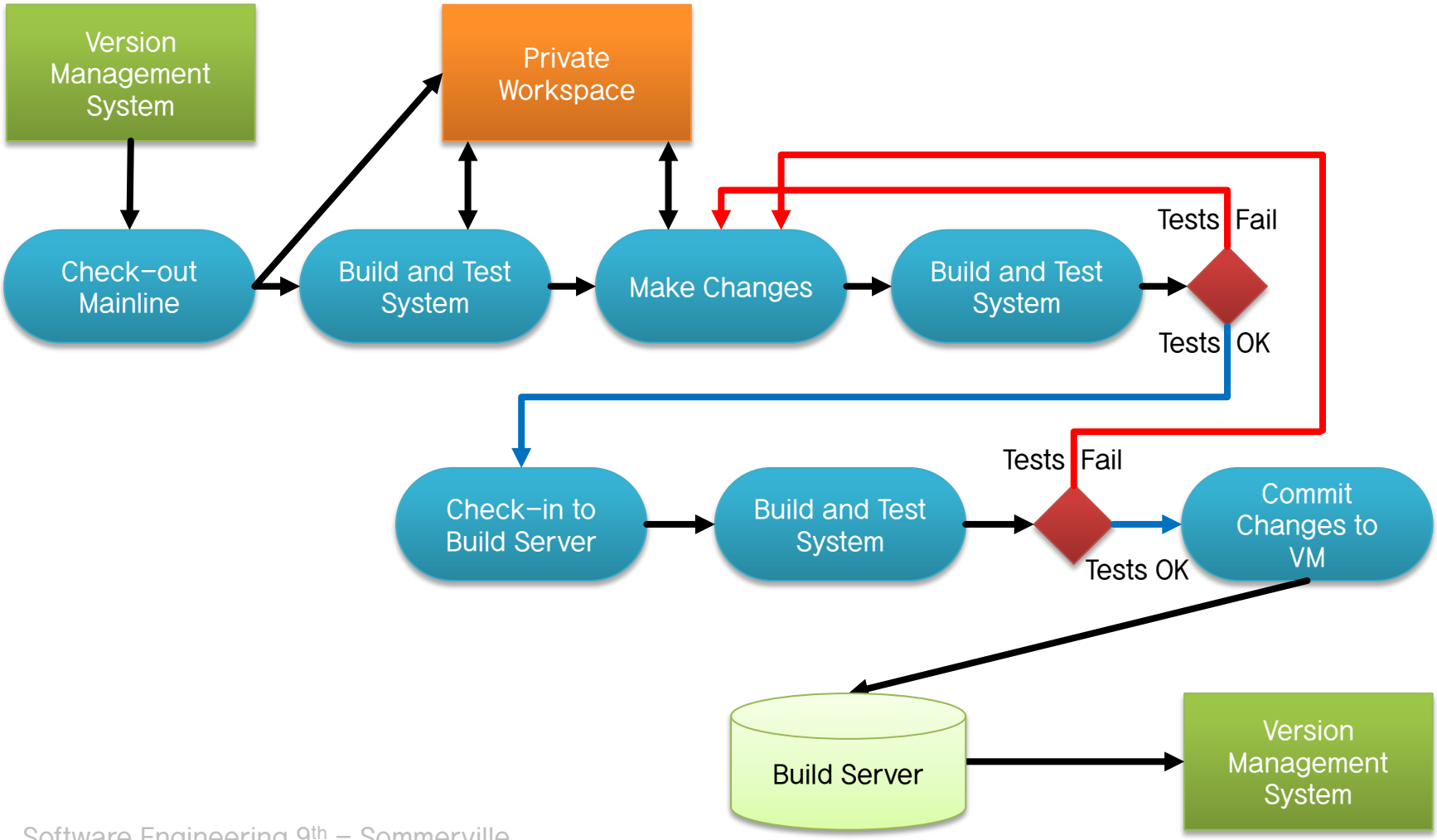
  + Continuous Test

Codeline and Baseline





CI Process


Version Management System
Private Workspace
Check-out Mainline
Build and Test System
Make Changes
Build and Test System
Tests Fail
Tests OK
Check-in to Build Server
Build and Test System
Tests Fail
Tests OK
Commit Changes to VM
Build Server
Version Management System

COMPONENTS?

Tool	Name
Project Management	OpenProj, GanttProject, Mylyn, Hsarepoint, ……
Quality Management	Splint, PMD, JDepend, Checkstyle, ……
Requirement Analysis & Management	Jfeature, JRequisite, ……
Design & Modeling	StarUML, VioletUMLEditor, ArgoUML, ……
Configuration Management & Version Control	Trac, Jenkins (Hudson), Maven, CVS, SVN, Bugzillia, Mantis, TortoiseCVS, Mercurial, Cruise Control, Git, Ant, RedMind, Berkshelf, Perforce, Nagios, Sensu, Vagrant, Foodcritic, ……
Implementation	Eclipse, Valgrind, Netbeans, ……
Testing	TestLink, CPPUnit, OProfile, HttpUnit, EMMA, Jmeter, Nunit, SoapUI, FitNesse, ……
Static Analysis	Coverity, dart, sparrow, cppcheclipse, Resort, sonar, HP Fortify Static Code Analyzer, Checkmarx CxSuite, Syhunt Sandcat, Parasoft {Jtest, dotTEST, …), VeraCode Static, Coverity Static Analysis,……

ADVANTAGES! (Wikipedia)

- When unit tests fail or a bug emerges, developers might revert the codebase to a bug-free state, without wasting time debugging
- Developers detect and fix integration problems continuously — avoiding last-minute chaos at release dates, (when everyone tries to check in their slightly incompatible versions).
- Early warning of broken/incompatible code
- Early warning of conflicting changes
- Immediate unit testing of all changes
- Constant availability of a "current" build for testing, demo, or release purposes
- Immediate feedback to developers on the quality, functionality, or system-wide impact of code they are writing
- Frequent code check-in pushes developers to create modular, less complex code [citation needed]
- Metrics generated from automated testing and CI (such as metrics for code coverage, code complexity, and features complete) focus developers on developing functional, quality code, and help develop momentum in a team
- .....

DISADVANTAGES? (Wikipedia)

- Initial setup time required
- Well-developed test-suite required to achieve automated testing advantages

2014 Software Verification

7 Team Practices

- TP#1(03.19) : Junit , Eclipse , 정적분석도구 3개
- TP#2(04.02) : 빌드 환경
- TP#3(04.16) : Mantis , SVN & CTIP
- TP#4(05.01) : 각 팀별로 SMA에게 CTIP 환경 전수
- TP#5(05.14) : 1st System Testing
- TP#6(05.28) : 2nd System Testing & Static Analysis
- TP#7(06.11) : Final Presentation – English

Team Projects 2014 Software Verification

3학년 "소프트웨어 모델링 및 분석" 수업과 연계

- 3학년 수업의 개발 결과물에 대해서 시스템 테스팅을 수행한 후 결과를 Mantis를 사용하여 공유합니다.
- 3학년 수업에서 단위 테스팅 및 관련 분석을 수행할 수 있도록 CTIP 환경을 제공합니다.
- 테스팅 결과에 대한 의견교환은 Mantis를 사용합니다.
- 테스팅을 위한 소스코드는 SVN을 통해 공유합니다.