LDTS

@ L.EIC

@rmaranhao

Today

- System Build
- Continuous Integration

System Build

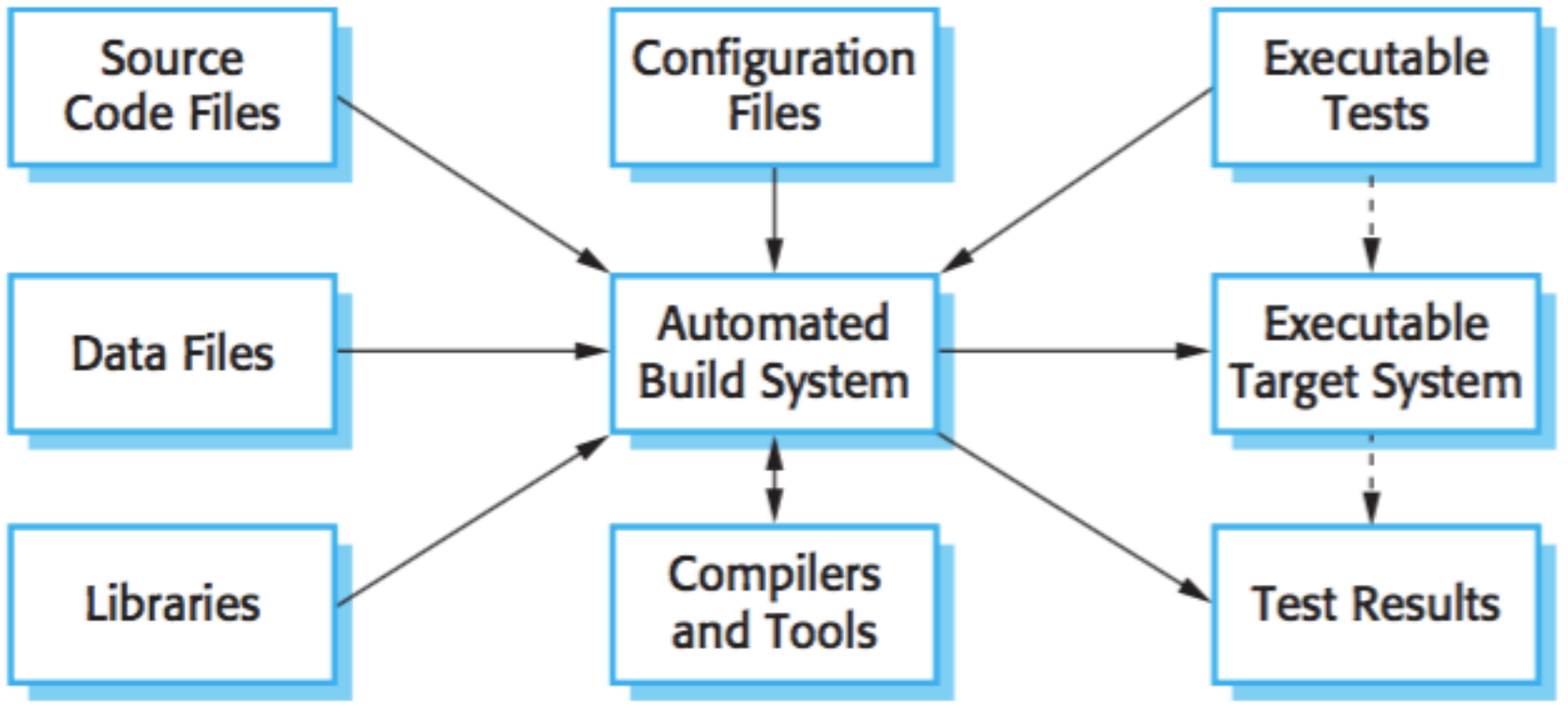
Building a system is a complex activity

Repetitive with variations

=>

It is error prone

From source files to executable files for test and/or deploy



(Sommerville, Fig 25.10)

Features of a build system

- Build script generation
- Version control system integration
- Minimal recompilation
- Executable system creation
- Test automation
- Reporting
- Documentation generation

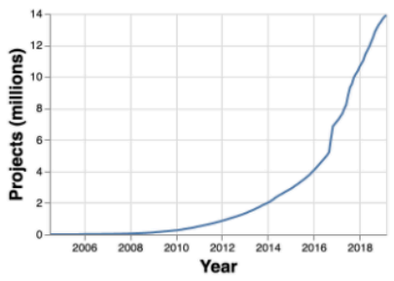
Conflicts between versions

compatible library versions

External repositories

contributors maven-ify their libraries

Indexed Artifacts (24.3M)



Popular Categories

Aspect Oriented
Actor Frameworks
Application Metrics
Build Tools
Bytecode Libraries
Command Line Parsers
Cache Implementations
Cloud Computing
Code Analyzers
Collections
Configuration Libraries
Core Utilities
Date and Time Utilities
Dependency Injection

Home » junit » junit » 4.13.2

JUnit » 4.13.2

JUnit is a unit testing framework for Java, created by Erich Gamma and Kent Beck.

License	EPL 1.0
Categories	Testing Frameworks
Organization	JUnit
HomePage	http://junit.org
Date	(Feb 13, 2021)
Files	jar (375 KB) View All
Repositories	Central
Used By	10,945 artifacts

Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

```
// https://mvnrepository.com/artifact/junit/junit
testImplementation group: 'junit', name: 'junit', version: '4.13.2'
```

Include comment with link to declaration

Home » junit » junit

https://mvnrepository.com/artifact/junit/junit

JUnit

JUnit is a unit testing framework for Java, created by Erich Gamma and Kent Beck.

License	EPL 1.0
Categories	Testing Frameworks
Tags	testing junit
Used By	65,412 artifacts

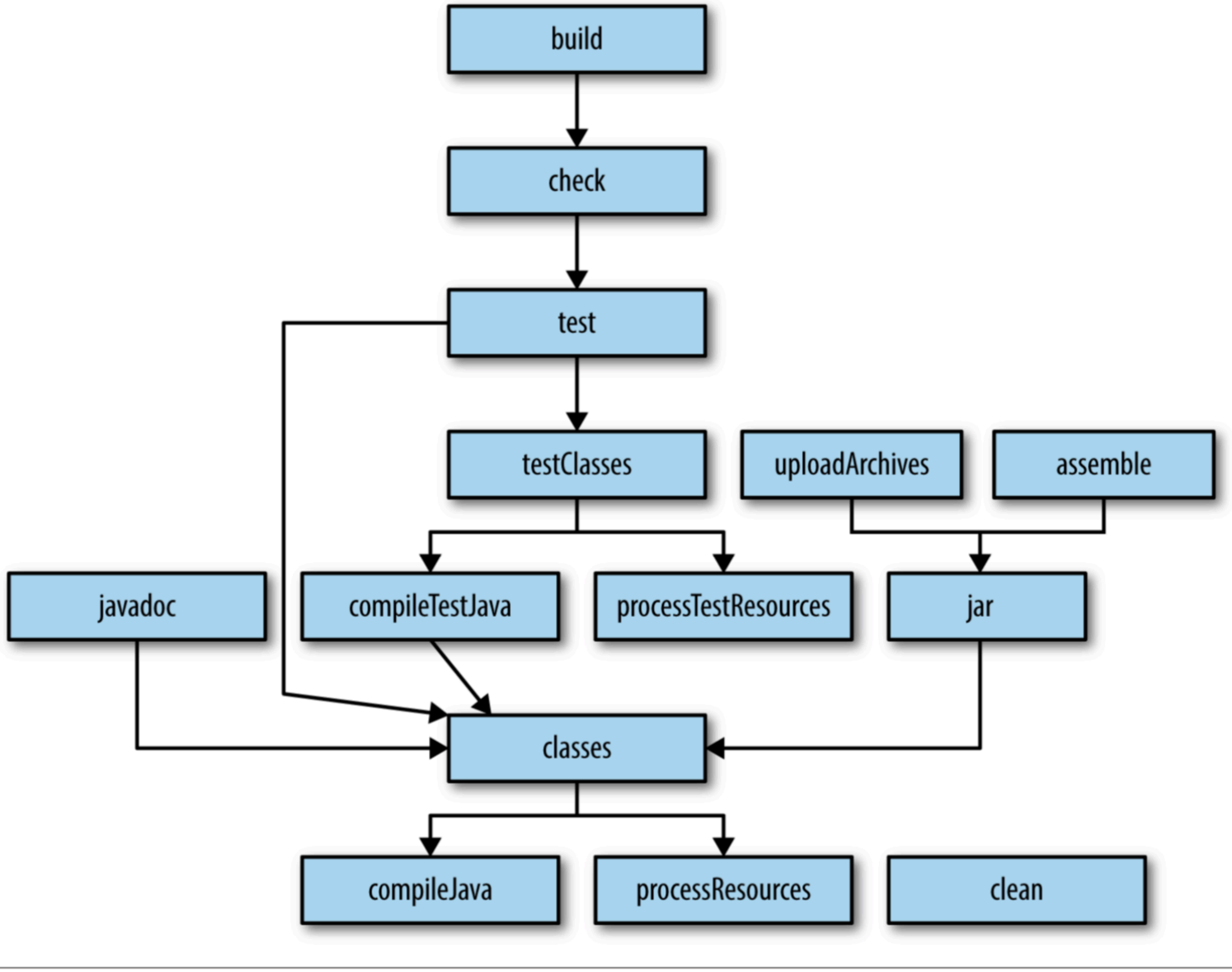
Central (24) | Redhat GA (3) | JBoss 3rd-party (1) | Alfresco Public (1)

	Version	Repository	Usages	Date
4.12.x	4.12	Central	26,423	(Dec, 2014)
	4.12-beta-3	Central	28	(Nov, 2014)
	4.12-beta-2	Central	31	(Sep, 2014)
	4.12-beta-1	Central	31	(Jul, 2014)
4.11.x	4.11	Central	20,211	(Nov, 2012)
	4.11-beta-1	Central	22	(Oct, 2012)
4.10.x	4.10	Central	7,936	(Sep, 2011)
4.9.x	4.9	Central	1,099	(Aug, 2011)
4.8.x	4.8.2	Central	4,998	(Oct, 2010)
	4.8.1	Central	4,579	(Feb, 2010)
	4.8	Central	241	(Oct, 2010)
4.7.x	4.7	Central	1,920	(Aug, 2009)
4.6.x	4.6	Central	720	(May, 2009)
4.5.x	4.5	Central	1,564	(Sep, 2008)
4.4.x	4.4	Central	2,274	(Aug, 2007)

It is repetitive

every time the system is built

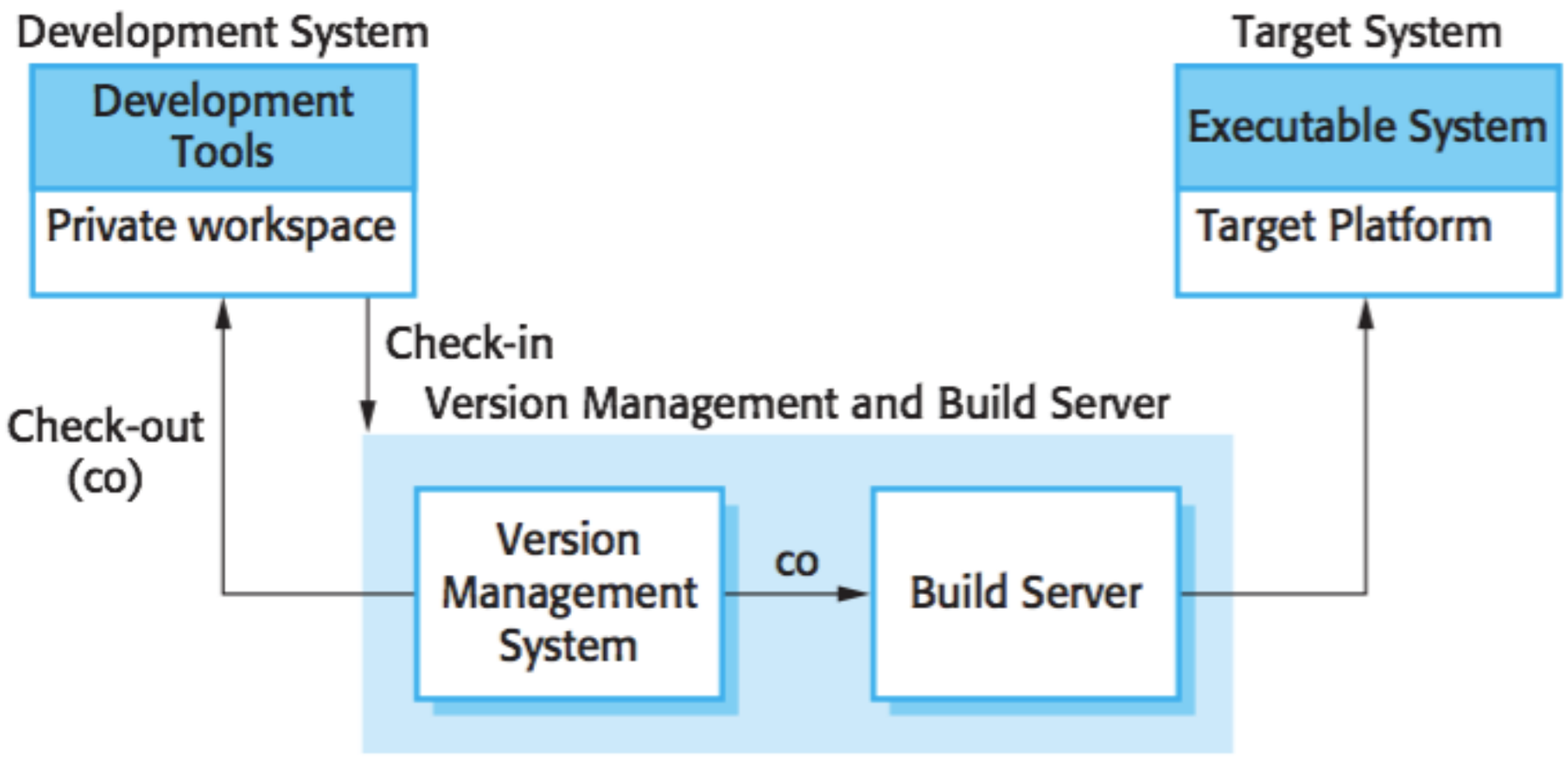
Gradle lifecycle


build
check
test
testClasses
uploadArchives
assemble
javadoc
compileTestJava
processTestResources
jar
classes
compileJava
processResources
clean

Can have several variations

build to compile,
build to run all tests,
build to deploy,
build the product for a client …

Build systems integrate with version management systems



(Sommerville, Fig 25.11)

Collaboration and Isolation

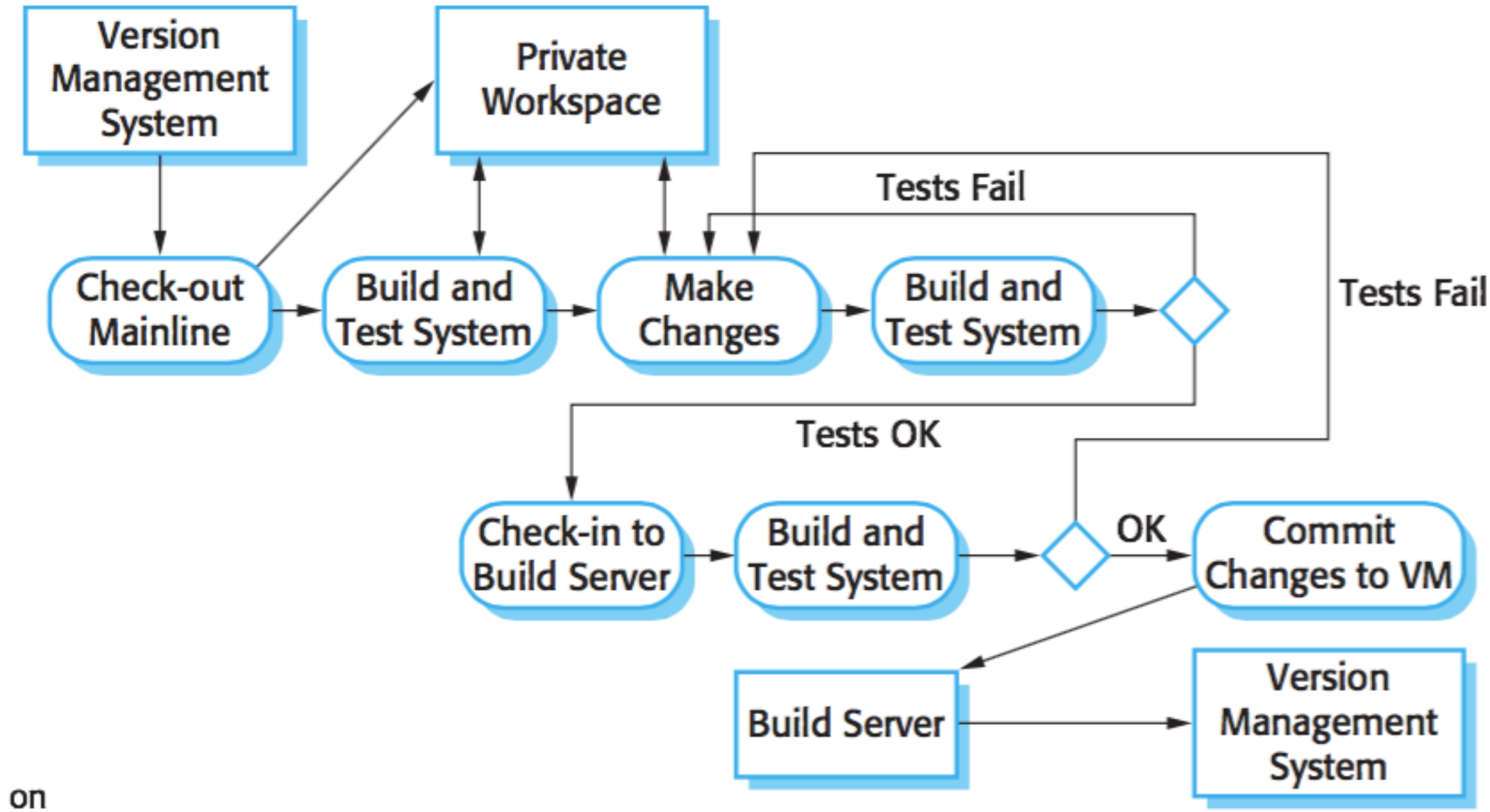
in the build process

Private build

for private development level of quality

Project build

for project development level of quality



on

(Sommerville, Fig 25.12)

Which steps you don't do in your project?

Continuous Integration

short cycles of isolation ⇔ collaboration

- Continuous integration may not be practical
  - system take a long time to build and test
  - development platform is different from the target platform

daily builds!