Build automation

Build automation Tools

- Automates braindraining repetitive work
  - compiling
  - testing
  - creating artifacts

- Ensures automatic repeatable builds!

- List_of_build_automation_software
  - Make
  - Ant
  - Maven
  - npm
  - Gradle
  - SBT
  - …

- Correct target platform
- Correct source encoding
- Correct dependencies incl versions
- Correct tools including versions
- Builds on every source platform
- ...

Build automation Tools

- Enabler for automated
  - Continuous Integration
  - Continuous Delivery
  - Continuous Deployment

- Monitors source repository
- Triggers on each commit
- Runs command (maven, bash, …)
- Keeps history
- Notifies failure/Success

- Examples
  - Bamboo (Atlassian)
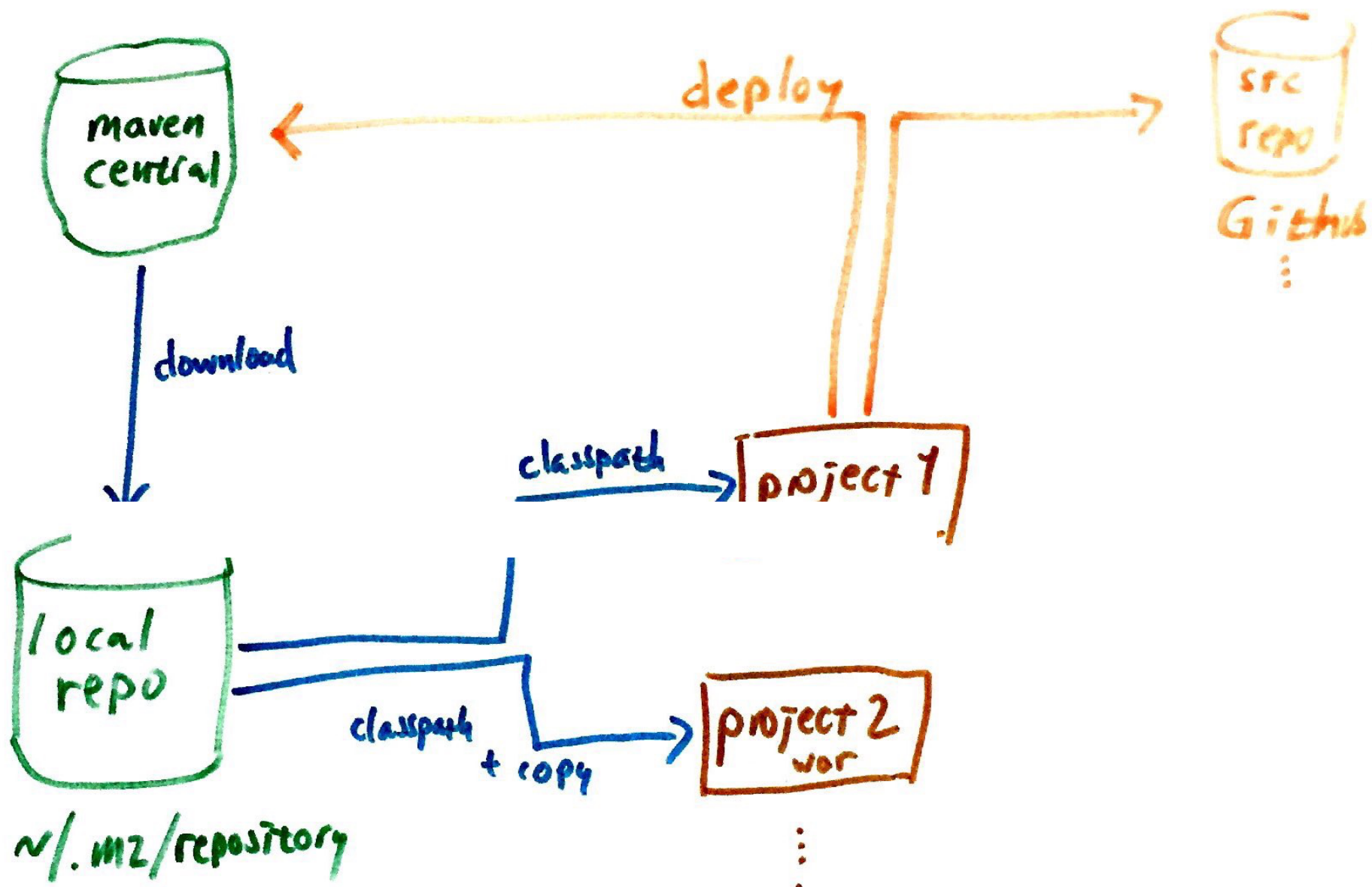 - Jenkins (OpenSource)
  - TeamCity (JetBrains)

Maven

- Enables repeatable builds
- Cleans up from earlier builds
- Puts dependencies on classpath
- Compiles all code
- Runs all tests
- Creates artifact, example project-jar-0.0.1-SNAPSHOT.jar
- Deploys maven repository and source repository
- Deploys to server
- IDE agnostic way of configuring project (Intellij, Eclipse, …)
- Filters resources
- Runs static analysis tools
  - Coverage reports

- And more!

Maven

- dependency management
- maven coordinates
- packaging
- build phases
- plugins
- scopes
- project structure
- project file - pom.xml
- commands

- ```
  <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.8.1</version>
  </dependency>
  ```

- https://mvnrepository.com/artifact/junit/junit/4.8.1
- ~/.m2/repository/junit/junit/4.8.1

- packaging
  - pom
  - jar - default
  - war
  - ear

- Default lifecycle
  - validate   - validate project
  - compile   - compile sources
  - test   - run tests
  - package   - package in distributable format
  - verify   - check quality criteria
  - install   - copy into local repository
  - deploy   - copy to remote repository

- Clean lifecycle
  - clean - removes all temporary files (target)

- plugin
  - goal —> phase

- maven-compiler-plugin
  - compiler:compile —> compile

- maven-resources-plugin
  - resources:resources —> process-resources

- maven-surefire-plugin
  - surefire:test —> test

- Each dependency has a scope, compile by default

- compile - transitive, compile, test, run phases
- provided - non-transitive, compile, test phases
- runtime - transitive, run, test phases
- test - non-transitive, test phases

- project-root/
  - pom.xml
  - src/main/java
  - src/main/resources
  - src/test/java
  - src/test/resources
  - target

- cd project; find src

- Import into eclipse

- mvn clean
- mvn test
- mvn package
- mvn install
- mvn exec:java
- mvn deploy
- mvn release:prepare; mvn release:perform

Film inför 24/4

The Magic Tricks of Testing by Sandi Metz 32 min
https://www.youtube.com/watch?v=URSWYvyc42M

För tips om hur man undviker sköra tester