

Szoftverellenőrzési technikák (vimim148)

Automatikus teszt futtatás, build keretrendszerek

Micskei Zoltán, Monostori Dénes

<http://www.inf.mit.bme.hu/>

Fordítás, tesztelés, kiadás nagy projekteknél

- Sokáig tartó tesztelés
 - Eclipse: 6 óra 40 perc (2010 márciusában)
- Sok fejlesztő
- Sok változat
 - Classic, JEE, Modeling...
- Sokféle platform
 1. Windows, Linux, Mac OSX, Solaris...
 2. Win32, GTK, Motif, Carbon...
 3. x86, x86_64, sparc...

Automatikus tesztelés

- Teszt futtatás, kiértékelés automatizálása
- Manuális vagy automatikus?
 - Nehézség
 - Pl. GUI, CD írás, rajzolás
 - Tesztelés élethossza
 - Meddig kell a teszt, milyen gyakran
 - Pontosság
 - Hibás pozitív (false positive)

3

Automatikus tesztelés tipikus lépései

Setup

- Legfrissebb verzió telepítése
- Különböző platform, OS, böngésző...
- Virtuális gépek, „Lab manager” programok

Execution

- Egyszerű script / xUnit / keretrendszer
- Naplózás

Analysis

- Teszt kiértékelése
- Sokszor nem triviális

Reporting

- Tesztek ezrei esetén nem elegendő a naplófájlok
- Összesítő információk

Cleanup

- Ismert, tiszta állapotba visszaállítás
- Cél: tesztek ne befolyásolják egymás futását

Help

- Teszt kód is ugyanolyan kód, azt is dokumentálni kell
- Sokszor a teszt kód hosszabb, mint az éles

4

Naplózás

- Források változásai
- Figyelmeztetések
- Tesztek
 - ID, név
 - Környezet
 - SUT (System Under Test) információk
 - Verzió
 - Beállítások
 - Nyelv
 - ...
 - Eredmények
- ...

5

Teszt futtatás – nagy léptékben

- Nagy projektek esetén (OS, böngésző, IDE...)
- Tesztek futtatása 10-100-1000 gépen
- Tipikus felállítás
 - Teszt vezérlő
 - Teszt adattár (kód, log, jelentések)
 - Ágens a tesztelő gépeken
- Pl.: Rational, Visual Studio, saját megoldások

6

Tartalom

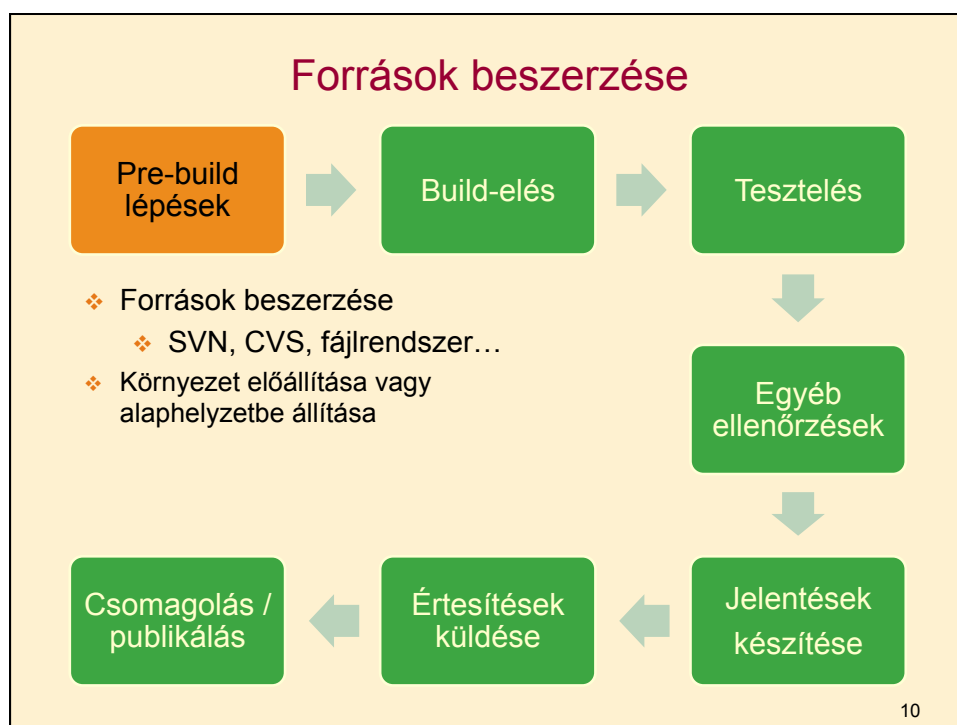
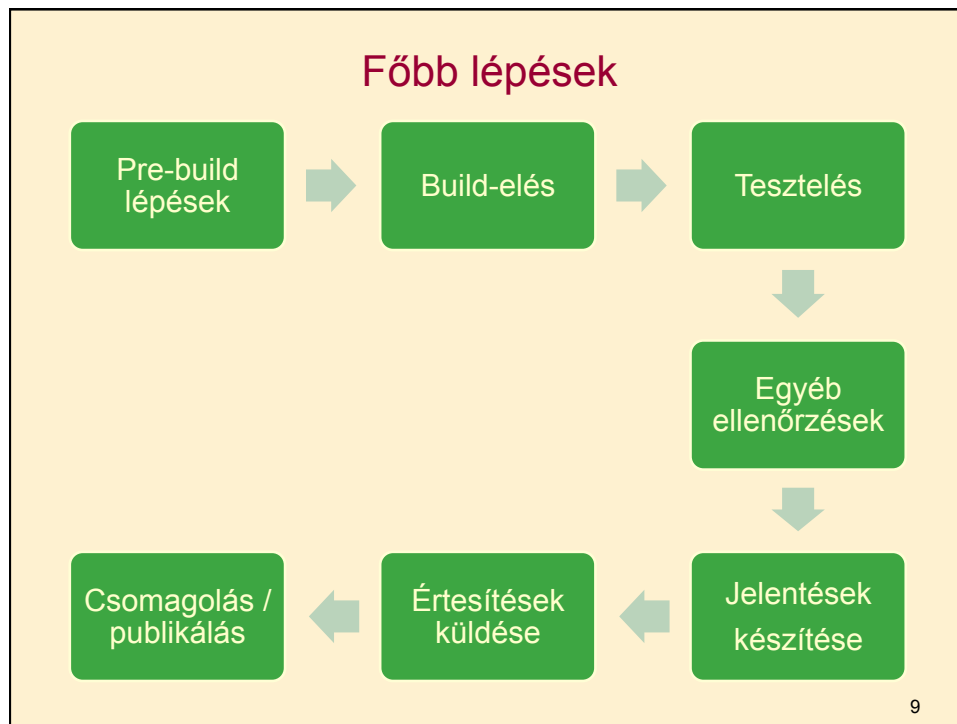
- Tesztelés automatizálása
- **Build folyamat**
- Build keretrendszerek

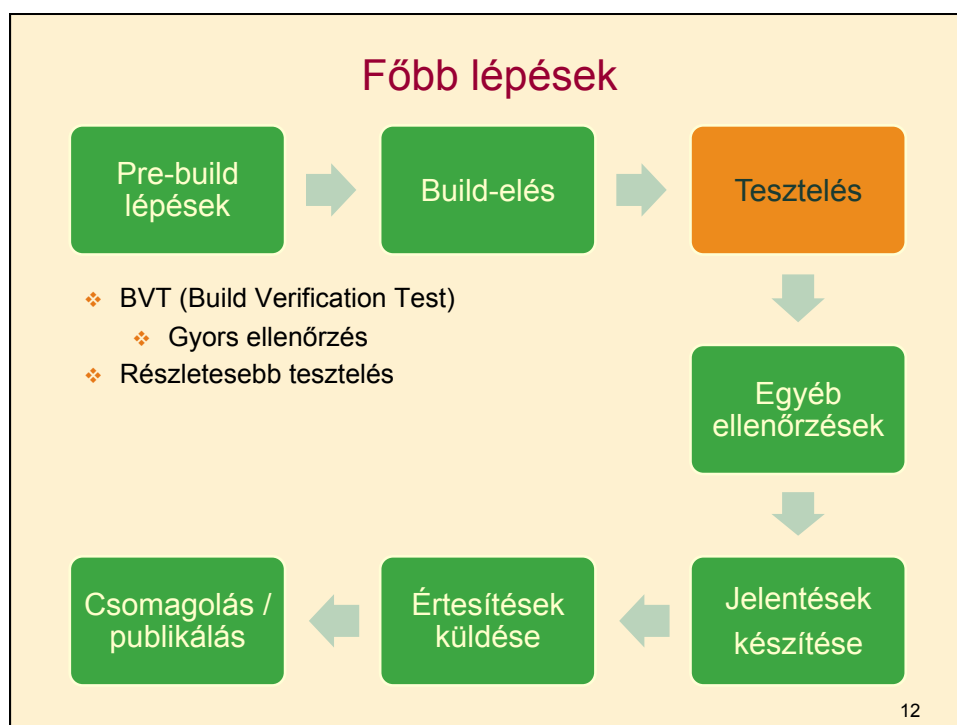
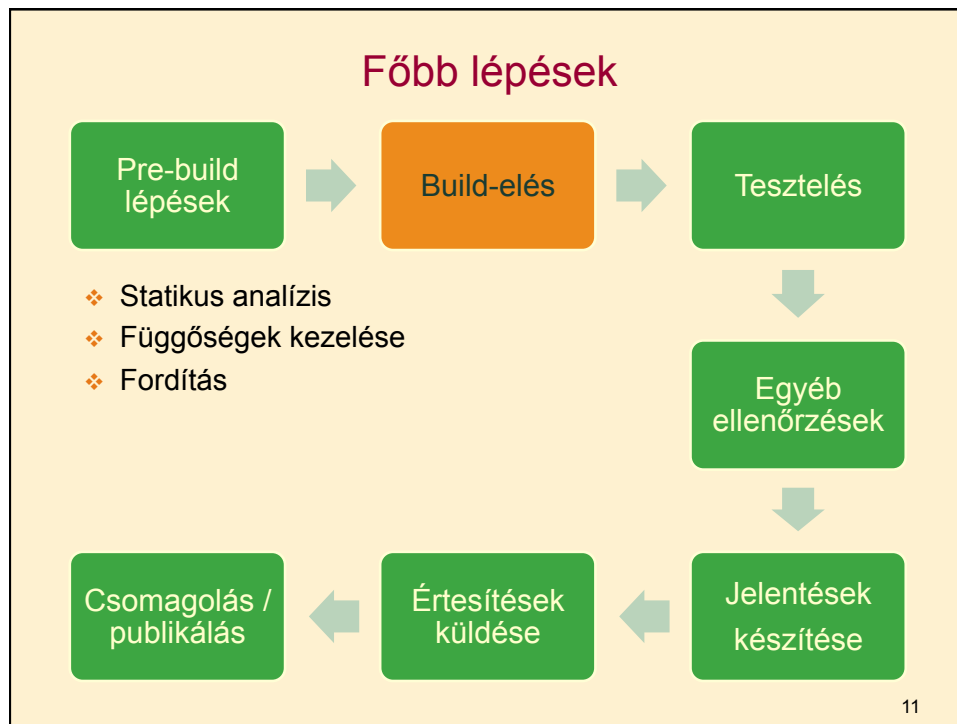
7

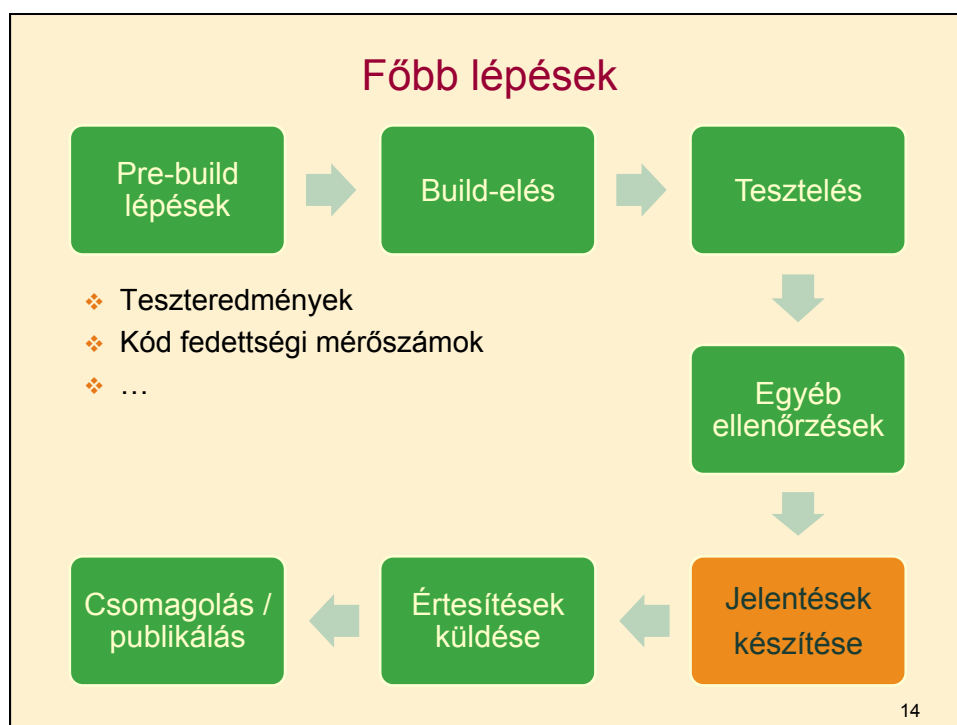
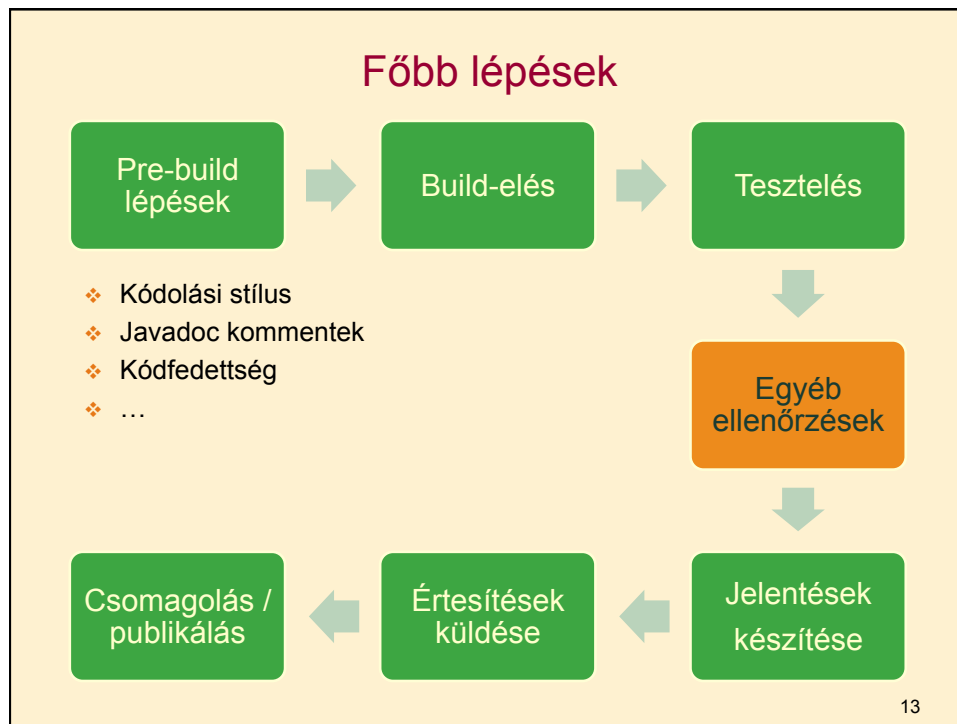
Központi build folyamat

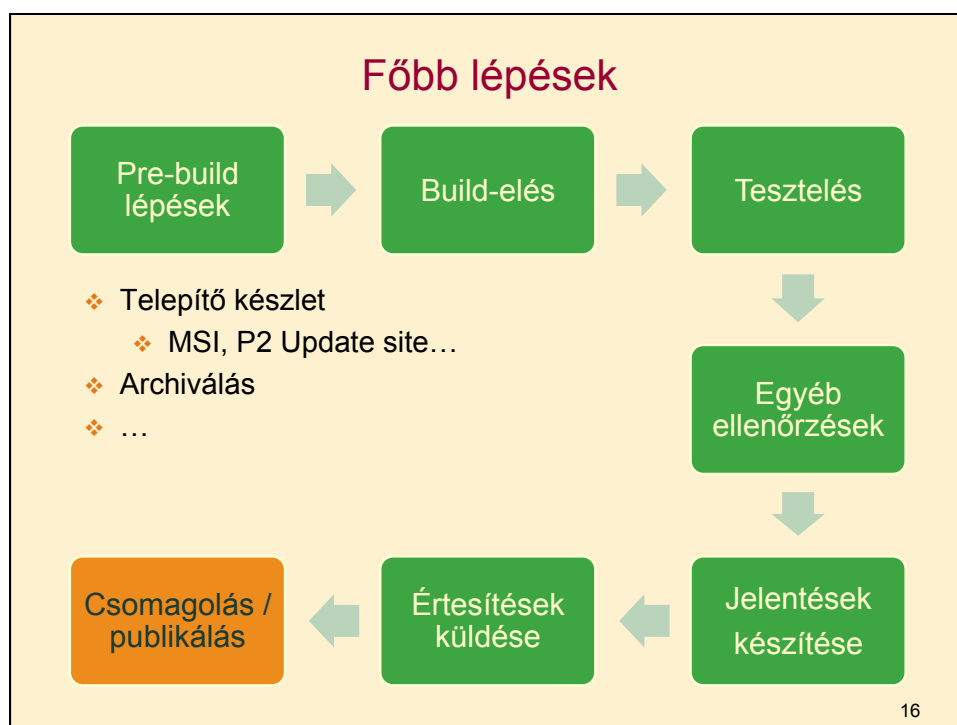
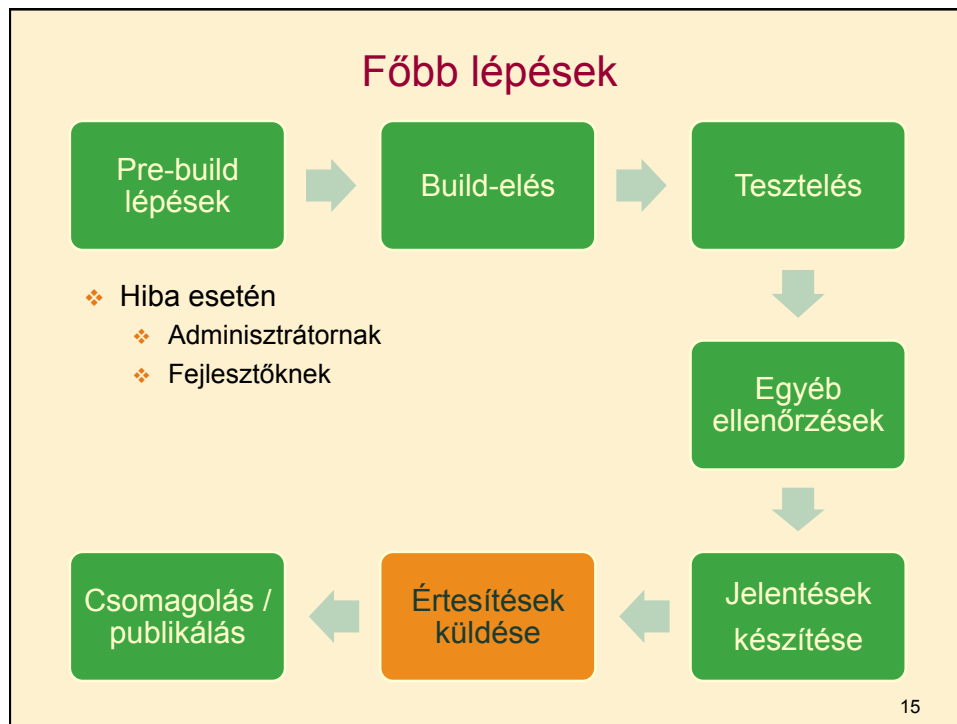
- Bonyolult, összetett folyamat
- Cél:
 - Futtatás egyetlen, **központi helyen**
 - Erőforrásigény
 - **Automatikusan**
 - Nem felejtődik el
 - GUI nélkül, **parancssorból**

8









Tartalom

- Tesztelés automatizálása
- **Build folyamat**
 - Build végrehajtó motorok
- Build keretrendszerek

17

Build végrehajtó motorok

- Make
 - C/C++
- Apache Ant
 - Make fájl Java-hoz, XML alapokon
- Apache Maven
 - Egységes forrás letöltés és fordítás
 - Funkcionalításában hasonlít az Ant-hoz
- ...

18

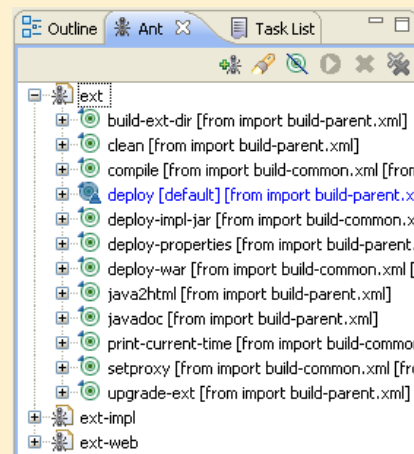
Ant

- Java library és parancssori eszköz
- Rugalmas, bővíthető
- Fő felhasználási terület:
 - Java alkalmazások build-elése

19

Ant alapfogalmak

- Project
 - Build fájlként egy
- Target
 - Végrehajtandó taszkok egy halmaza
 - 1..*
 - **Egymástól függhetnek**
 - Pl. compile, deploy
- Task
 - Végrehajtható kód
 - Pl. javac, copy, junit, exec, signjar, mail...



20

További elemek

- Név—érték párok (properties)

```
<property name="build" location="build"/>
```

```
<target name="init">
```

```
  <mkdir dir="${build}"/>
```

```
</target>
```

- Útvonalak, classpath

```
<classpath>
```

```
  <pathelement path="${classpath}"/>
```

```
  <pathelement location="lib/helper.jar"/>
```

```
</classpath>
```

- Bármely projektelemnek lehet ID-ja

– → Minden hivatkozható

21

Előkészületek

- Szükséges:

– junit.jar

– ant-junit.jar

- Alapértelmezett helye: ANT_HOME/lib

- junit.jar megadása:

– ANT_HOME/lib könyvtárba másolással, vagy

– -lib argumentummal, vagy

– <junit> taszk <classpath> elemében

22

Példa

```

<project default="test" >
  <path id="classpath.test">
    <pathelement location="lib/junit.jar" />
    <pathelement location="${build}" />
  </path>

  ...

  <target name="compile-test">
    <javac srcdir="${tst-dir}" >
      <classpath refid="classpath.test"/>
    </javac>
  </target>

  ...

```

23

Példa (folytatás)

```

...
<target name="test" depends="compile-test" >
  <junit printsummary="yes" haltonfailure="yes">
    <classpath refid="classpath.test" />
    <formatter type="plain" />
    <test name="hu.optxware.junitcourse.
      bookstore.book.test.BMListTest"
      haltonfailure="no" outfile="result"
    >
      <formattertype="xml"/>
    </test>
  </junit>
</target>

```

24

Maven

- **Leíró**
 - pom.xml: projekt modell
 - Archetípus: minta
 - Eltérések felsorolása a mintától
- **Fordítás**
 - Megnevezünk egy célt (pl. teszt, csomagolás)
 - Végignézi az összes szükséges fájlist

25

Példa: JUnit teszt futtatás Mavenrel

- **Projekt struktúra:**
- **my-app**
 - pom.xml
 - src
 - main
 - java
 - » com |
 - » mycompany
 - » app |
 - » App.java
 - test
 - java
 - com
 - » mycompany
 - » app
 - » AppTest.java

26

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://
maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Quick Start Archetype</name>
  <url>http://maven.apache.org</url>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.8.0</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>

```

27

Ant vs Maven

- Igazi “vallásháború”
- Ant
 - Minden kézben tartható
 - Egyedi projektnél hasznos
- Maven
 - “Convention over configuration”
 - Minden Maven projekt hasonló...
 - Függőségkezelés

28

Tartalom

- Tesztelés automatizálása
- Build folyamat
- **Build keretrendszerek**

29

Continuous Integration

- Gyakori agilis technika
- Céljai
 - **Minőség** növelése
 - Piacra kerülési idő csökkentése
- **Build szerver**
 - Automatikus integráció támogatása

30

Eszközök

- Apache Continuum (Java)
 - XML szerkesztés + webes UI
- CruiseControl (Java, .NET, Ruby)
 - XML szerkesztés
- Hudson (Java, de kiterjeszthető)
 - Webes UI
- TeamCity (Java, .NET, Ruby)
 - Fizetős
- ...

31

Hudson (Jenkins)

- Java szervlet alapú
 - Tetszőleges alkalmazás szerveren fut
- Plug-in alapú, **bővíthető**
- Frissítések keresése automatikus
- Gyorsan bele lehet tanulni
- Nem végez tényleges fordítást
 - Időzítés
 - Menedzselés
- Több folyamat, köztük akár **függőségekkel**

32

<https://hudson.eclipse.org/hudson/>

Hudson

Hudson

People
Építések Története
Projekt Kapcsolat
Fájl Ujilenyomat Ellenőrzése

Építési Sor

MWE-Languange-nightly-HEAD

Építés Futtató Állapota

Master

1 Idle

2 Building Xtext-nightly-HEAD #404

hudson-slave1

1 Idle

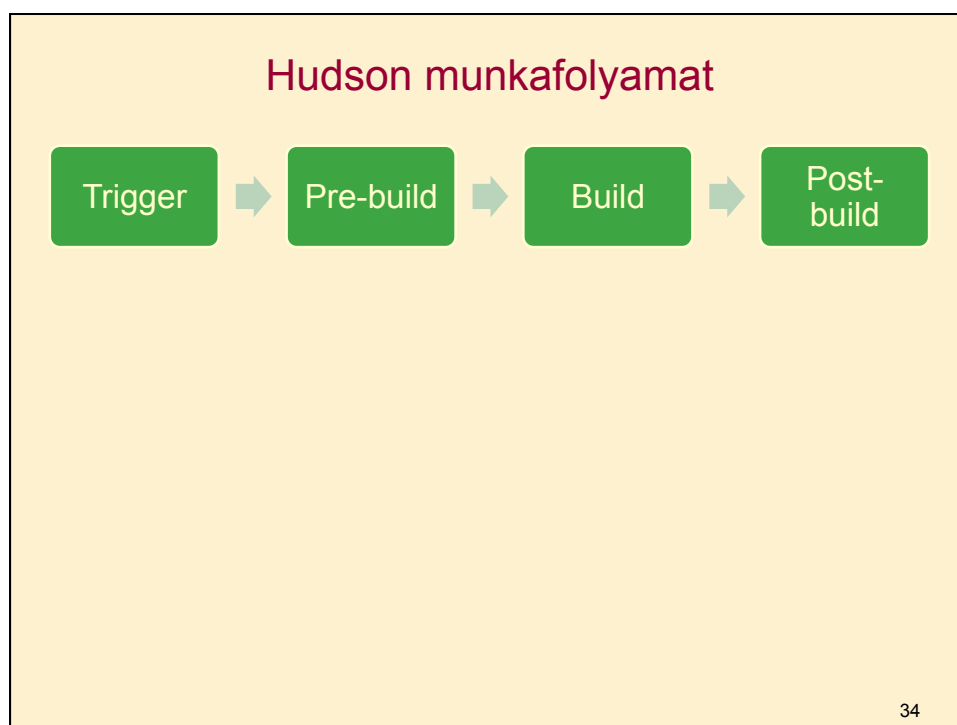
2 Idle

3 Building emf-cdo-integration #825

4 Idle

All	Amalgam	Athena CBI	Athena CBI (SVN)	Buckminster	Eclipse and Equinox	JWT	Jetty-RT	Mode
S	W	Job	I	Utolsó Sikeres				
		bpmel-0.5		4 days 5 hr (#29)				
		buckminster-egf-trunk-nightly		1 hr 49 min (#20)				
		buckminster-emft-ecoretools-0.10-nightly		N/A				
		buckminster-head		N/A				
		buckminster-maintenance		3 days 9 hr (#60)				
		buckminster-mdt-ocl-core-3.1-nightly		21 days (#57)				

33



Hudson munkafolyamat



- ❖ Kézi
- ❖ Időzített
- ❖ Verziókezelő rendszer változása
- ❖ Függő job befejeződése
- ❖ Egyéb (bővíthető)

35

Hudson munkafolyamat



- ❖ Opcionális
- ❖ Források beszerzése

36

Hudson munkafolyamat



- ❖ Tényleges fordítási lépések
- ❖ Beépített támogatás
 - ❖ Ant
 - ❖ Shell script
- ❖ Bővítéssel
 - ❖ Maven
 - ❖ **Buckminster**

37

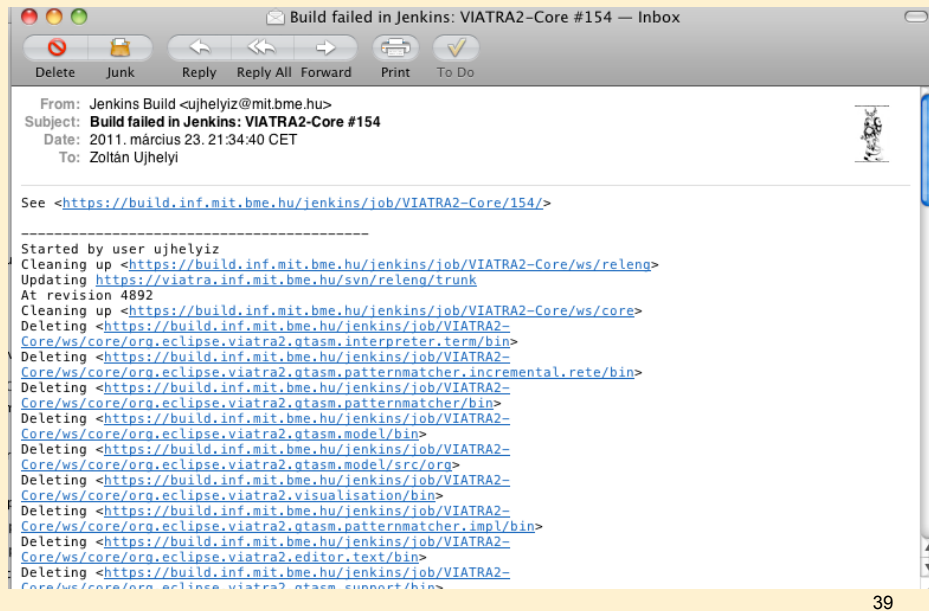
Hudson munkafolyamat



- ❖ Opcionális
- ❖ Archiválás
- ❖ Publikálás
- ❖ Független build-ek indítása
- ❖ Értesítések
- ❖ ...

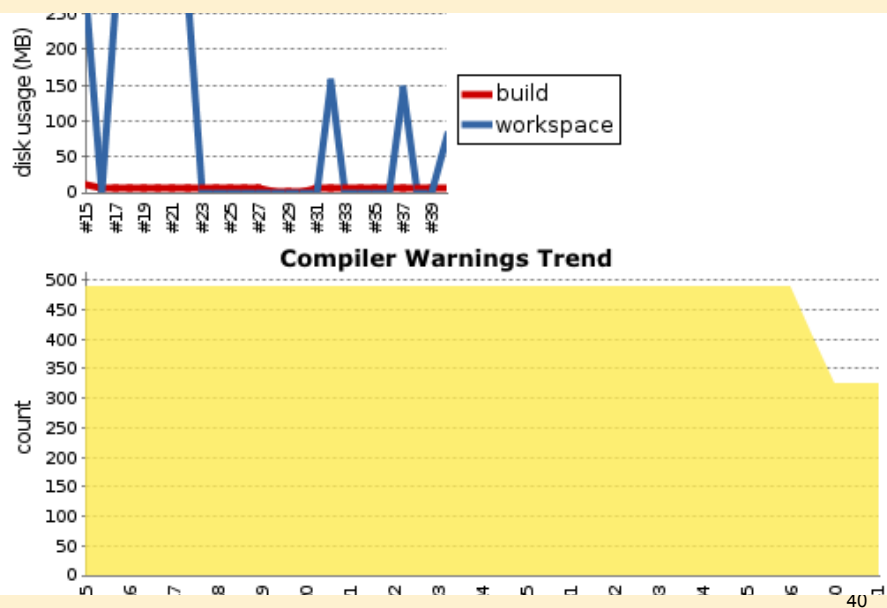
38

Blame mail

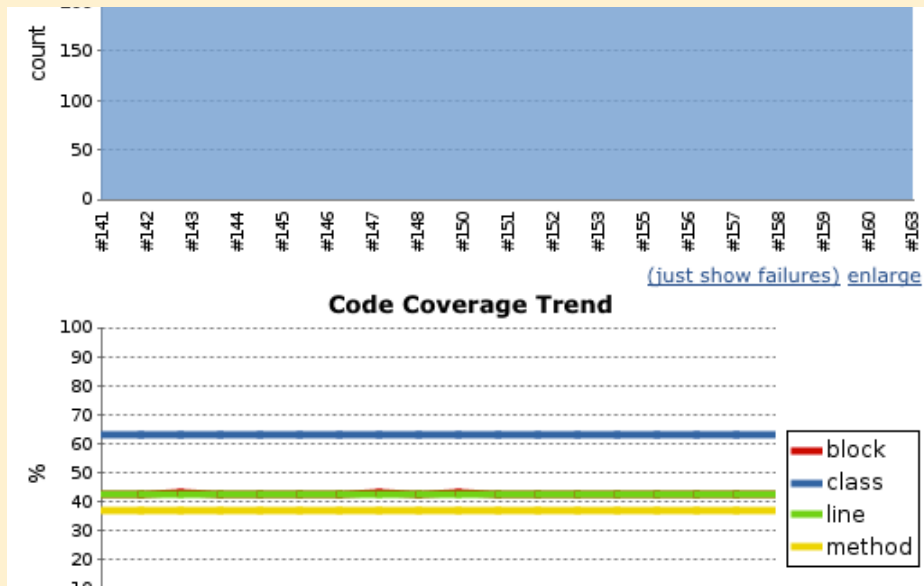


39

Metrikák, trendek

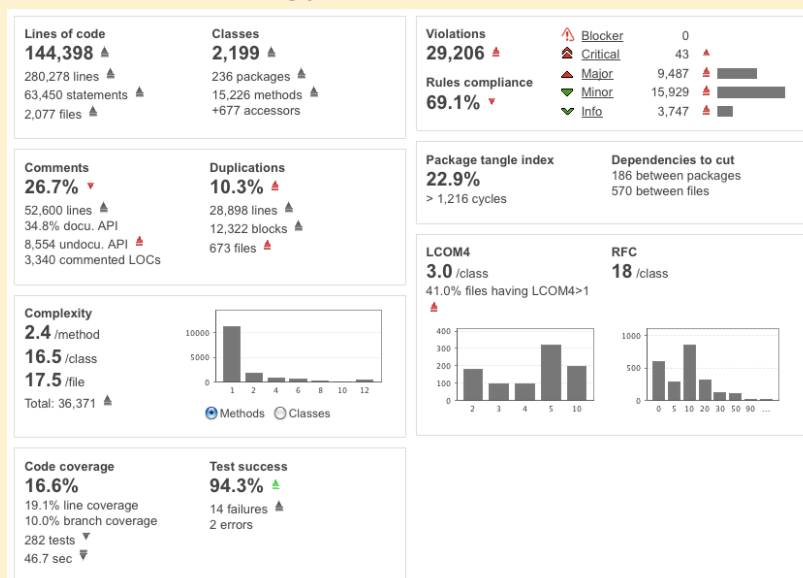


Kódfedettség trendek



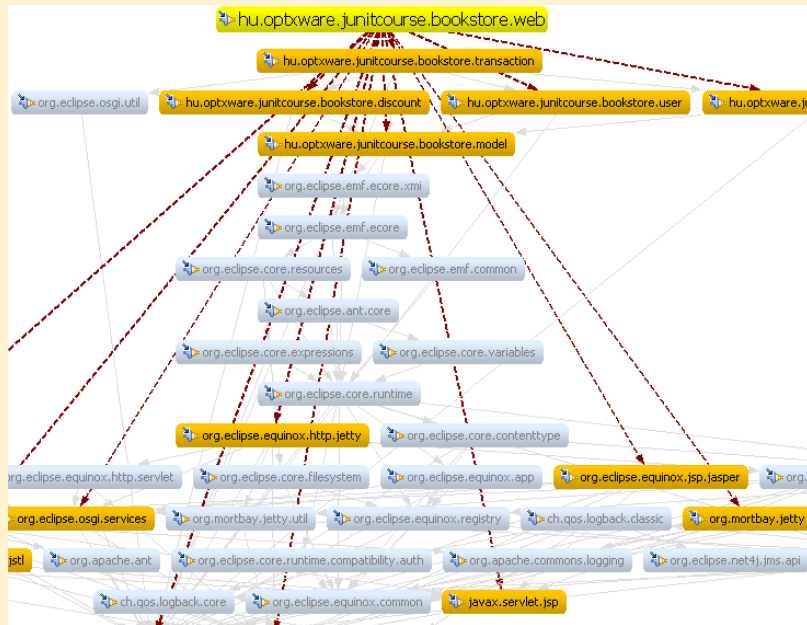
41

Egyéb metrikák



42

Probléma



43

Probléma (folytatás)



44

Probléma (folytatás)



45

Probléma (folytatás)



46

Probléma (folytatás)



47

Probléma (folytatás)



48

Probléma (folytatás)



Oh, hát persze, a D-nek csak az 1.2-es változata a jó, és másold be az E-t a plugins könyvtárba, majd...

49

OSGi függőségek kezelése

- Ant4Eclipse
 - PDE/Build kikerülése
- Pax, Tycho
 - Maven felkészítése OSGi függőségekre
- PDE headless build
 - Ant szkriptek generálásával
 - → Lényegében lehetetlen debug-olni
- **Buckminster**

(Mindegyik bonyolult!)

50

Buckminster

- Eclipse Tools Project
- Magas szintű eszköz
 - Meglévő eszközök felett fut
 - Ami Eclipse-ben fordítható, az Buckminsterrel is
 - **Leírók** segítségével
 - XML dokumentumok
 - Részben generáltak
 - Többihez szerkesztési támogatás
 - **Függőségek** kezelése

51

Felhasználási módok

- IDE támogatás
 - Leírók szerkesztése
 - Futtatás
- Headless mód
- Hudson plug-in

52

Képességek

- Forrás beszerzés
- Fordítás
 - PDE/Build, Ant, Maven
- Csomagolás
 - P2 update site
 - Target platform

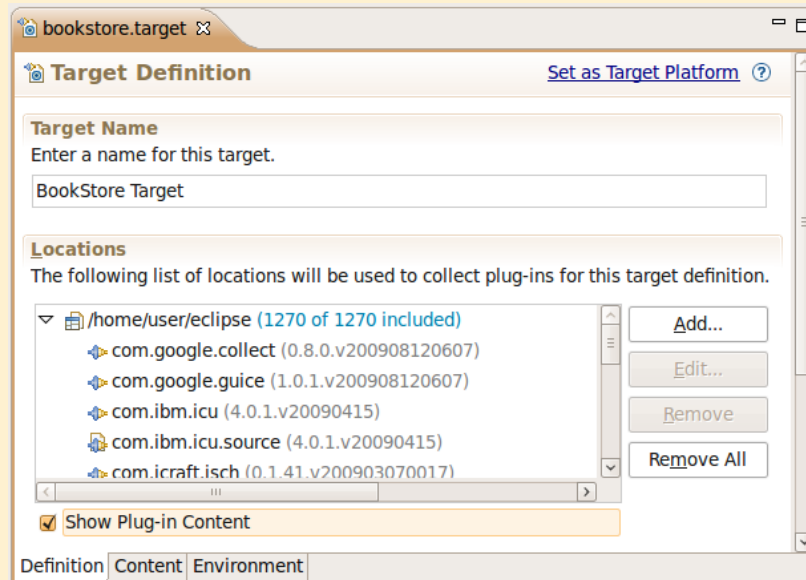
53

Kitérő: Target Platform

- Plug-in-ek egy halmaza, amit alapként használunk
 - Fordításhoz
 - Elég a hivatkozott interfész
 - Futtatáshoz
 - Kell a teljes implementáció
- Kapcsolódó beállítások
 - Célkörnyezet
 - JVM verzió
 - ...

54

Target Platform szerkesztő

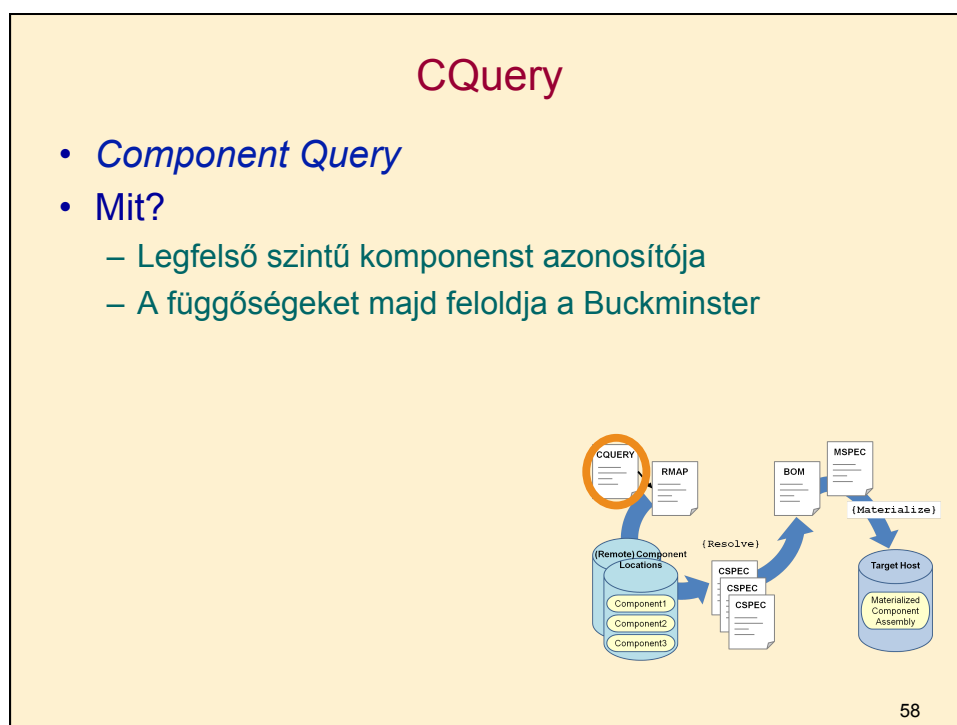
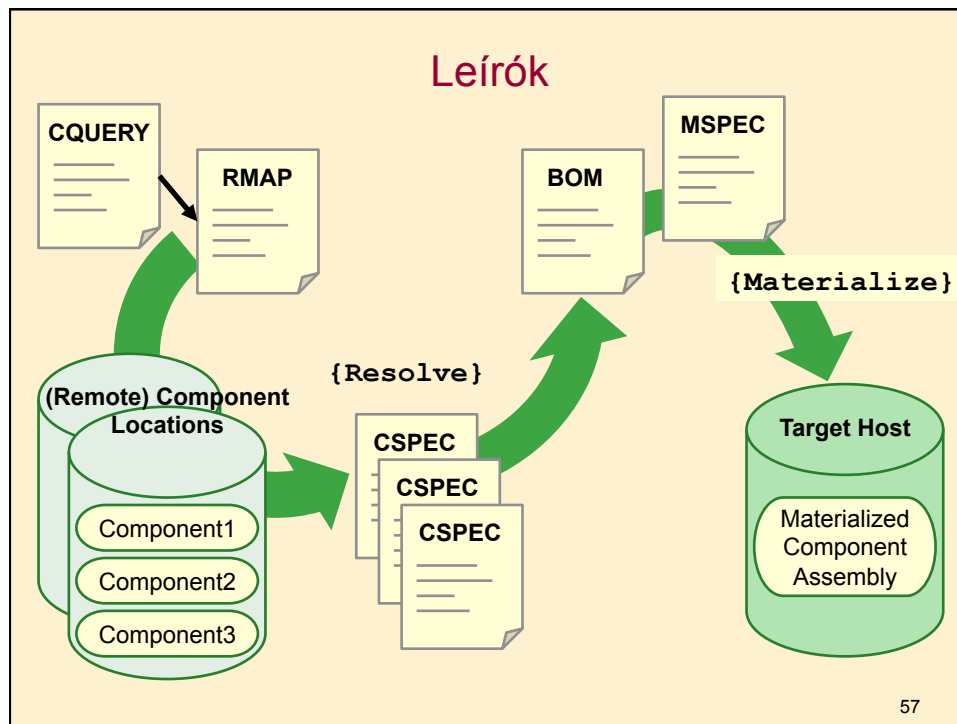


55

Komponens

- Egy egység (absztrakció)
 - Feature, plug-in...
 - Van neve, típusa, verziója
- Műveletek értelmezhetőek rajta
 - Néhány előre definiált (pl. site.p2, bundle.jar)
 - Saját műveletek

56

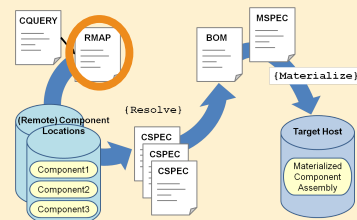


CQuery szerkesztő

59

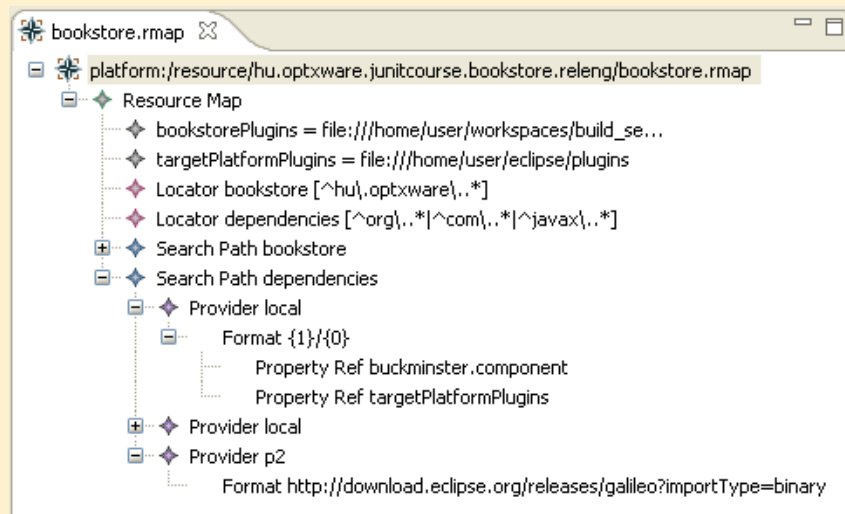
RMap

- *Resource Map*
- Honnan?
 - P2 update site
 - Lokális könyvtár
 - SVN, CVS, Git...
 - Maven
 - Target platform
 - Workspace
 - URL



60

RMap szerkesztő



61

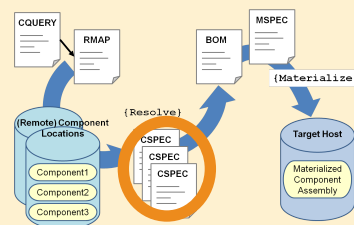
Opcionális paraméterek

- Forrás / bináris
- Branch, tag
- Akár komponensenként különböző repository-k
 - Release
 - Nightly

62

CSpec

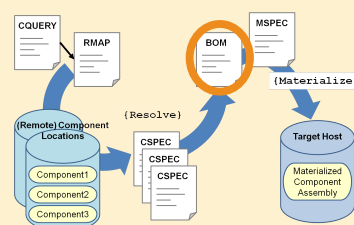
- *Component Specification*
- Generált
- Ebben szerepelnek pl. a komponensen elvégezhető műveletek is
- Saját kiegészítések:
 - *CSpeX (CSpec eXtension)*



63

BOM

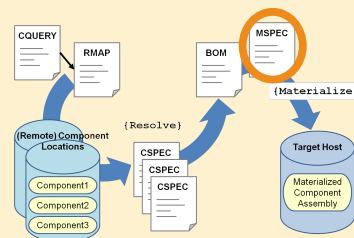
- *Bill Of Materials*
- Generált
- Konkrét helyek és teendők listája



64

MSpec

- *Materialization Specification*
- Mit hova tegyen?
 - Workspace
 - Target platform
 - Adott könyvtár
- Alapértelmezés 3.6 óta:
 - Források workspace-be
 - Binárisok target platformba



65

Buckminster - összefoglalás

- Összegyűjtötte a komponenseket
- Definiálta a műveleteket
- Innentől jöhet a tényleges build...

66

Összefoglalás

- **Teszt automatizálás**
 - Összetett folyamat
 - Sok részlépés
 - Külön-külön is automatizálható
- **Build folyamat**
 - Minimálisnál nagyobb projekt esetén „kötelező”
 - Jó eszköztámogatás
 - Nem triviális beállítani