Szoftverellenőrzési technikák (vimim148)

Automatikus teszt futtatás, build keretrendszerek

Ujhelyi Zoltán, 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…
- *“Shipping is hard, that’s why we do it 7 times a release.”*

Mozilla Firefox

- 17 platform
- 12 branch forrásnak
- 1200 build and teszt gép
  - Fordítási idő: 12.40 óra
  - Tesztelési idő: 54.48 óra
  - CPU időben: 2.79 nap (!)
  - Korábban release: 10 nap

Forrás: http://relengofthenerds.blogspot.com/2010/11/mozilla-versus-eclipse-build.html

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)

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 elegek 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

Naplózás

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

Teszt futtatás – nagy léptékben

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

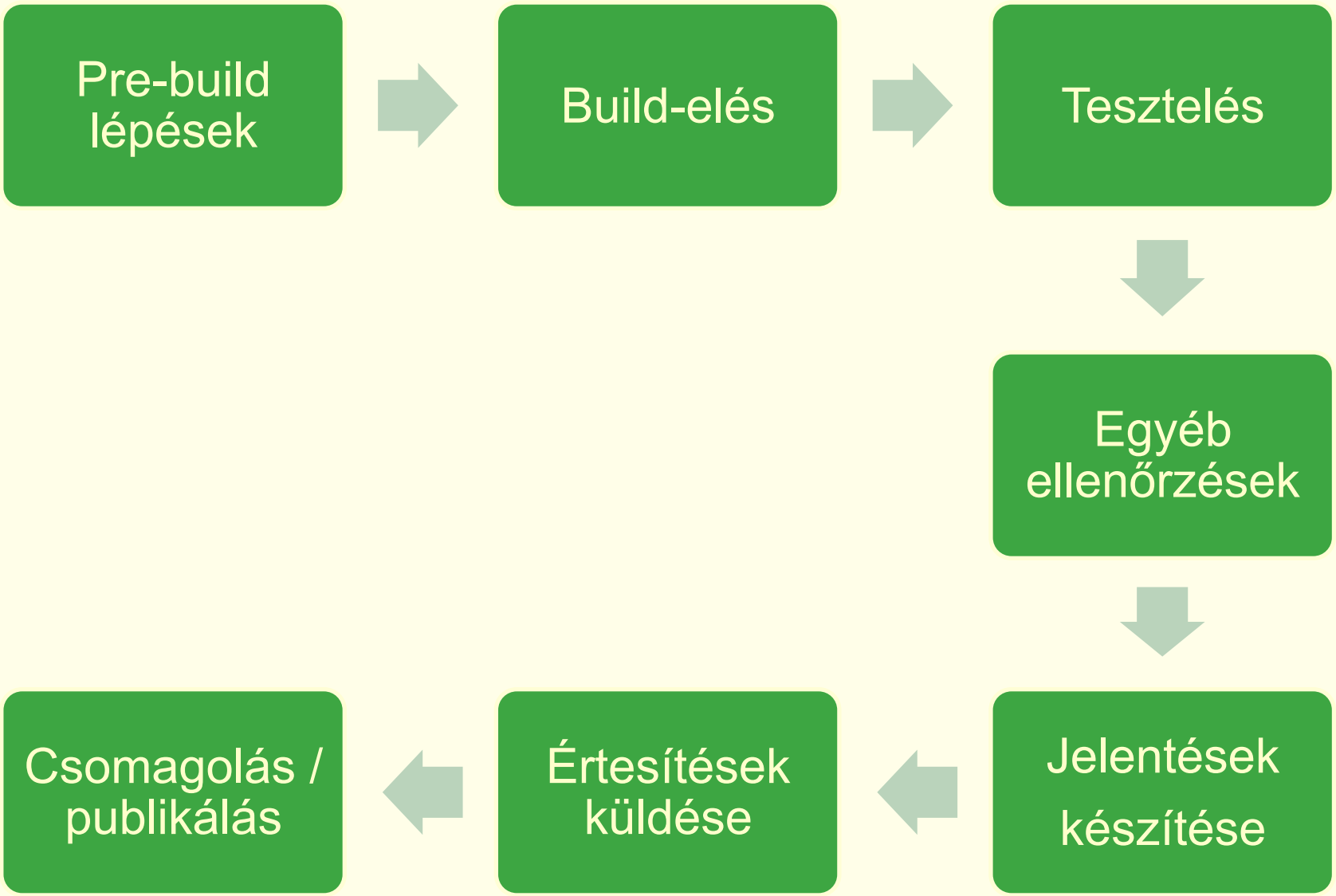
Tartalom

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

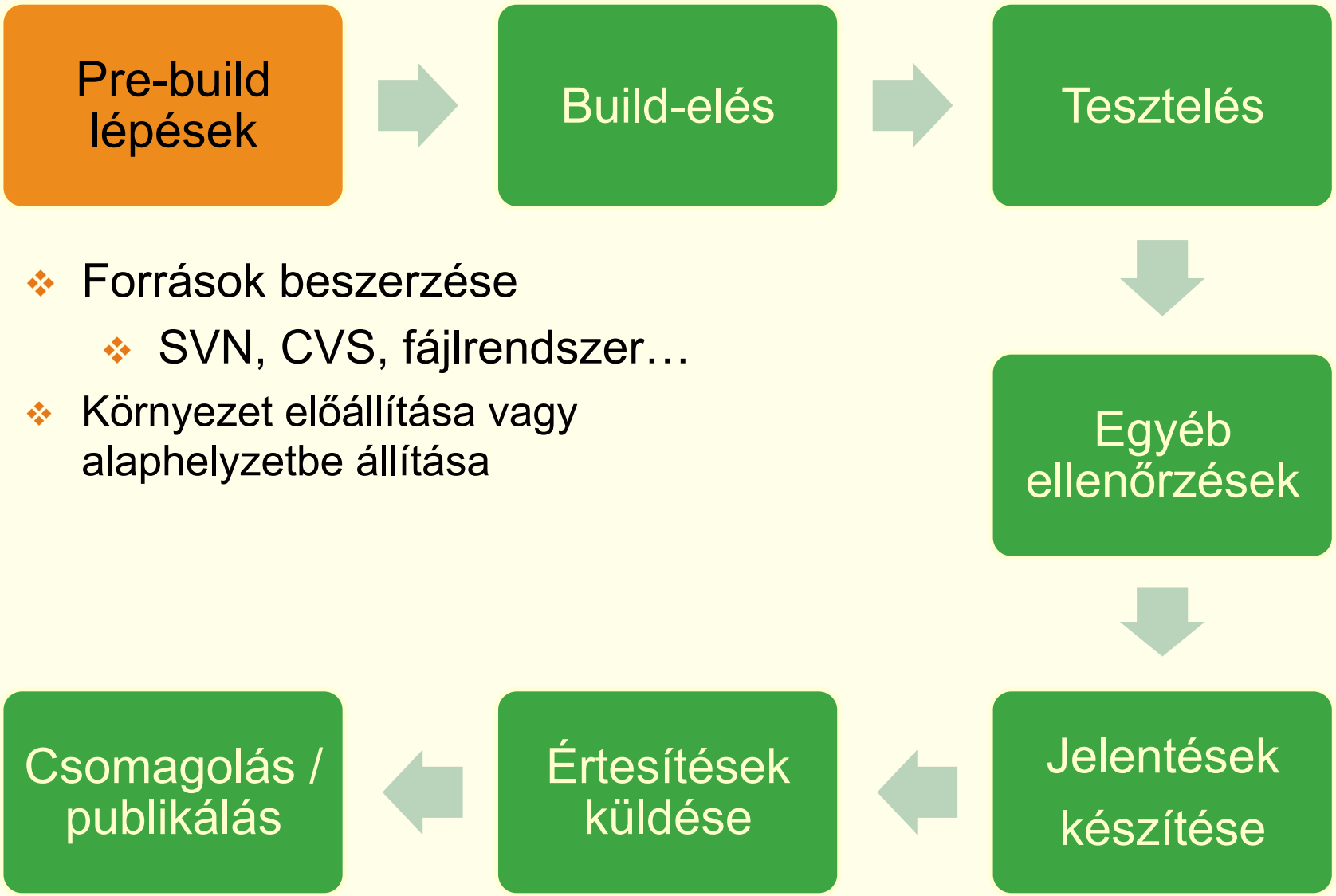
Központi build folyamat

- Bonyolult, összetett folyamat
- Cél:
  - Futtatás egyetlen, **központi helyen**
    - Erőforrásigény
  - **Automatikusan**
    - Nem felejtődik el
  - **Környezetfüggetlen**, fut
    - Fejlesztő gépén parancssorból (GUI nélkül)
    - Fejlesztőkörnyezetben!
    - Build szerveren

Főbb lépések

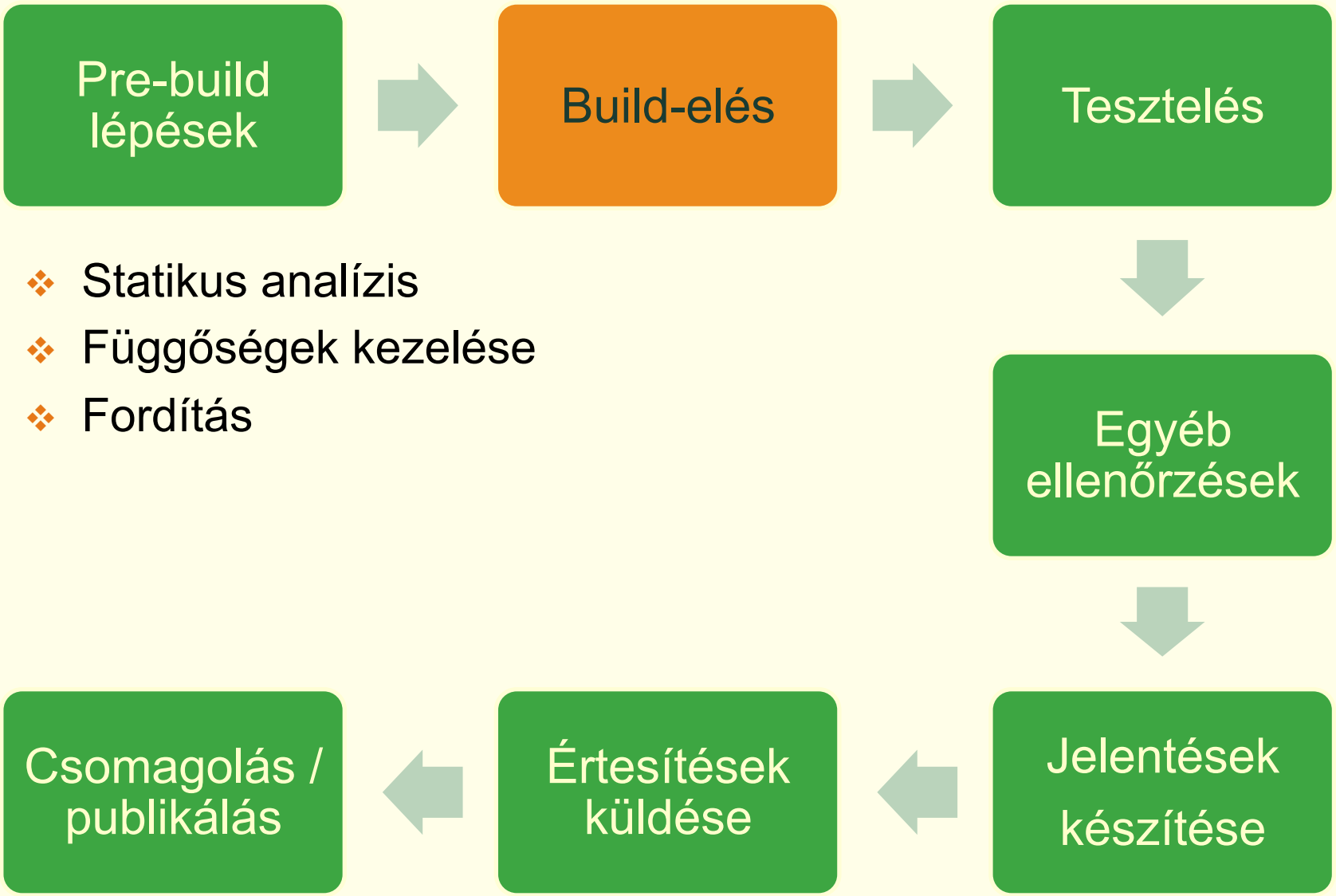


Források beszerzése



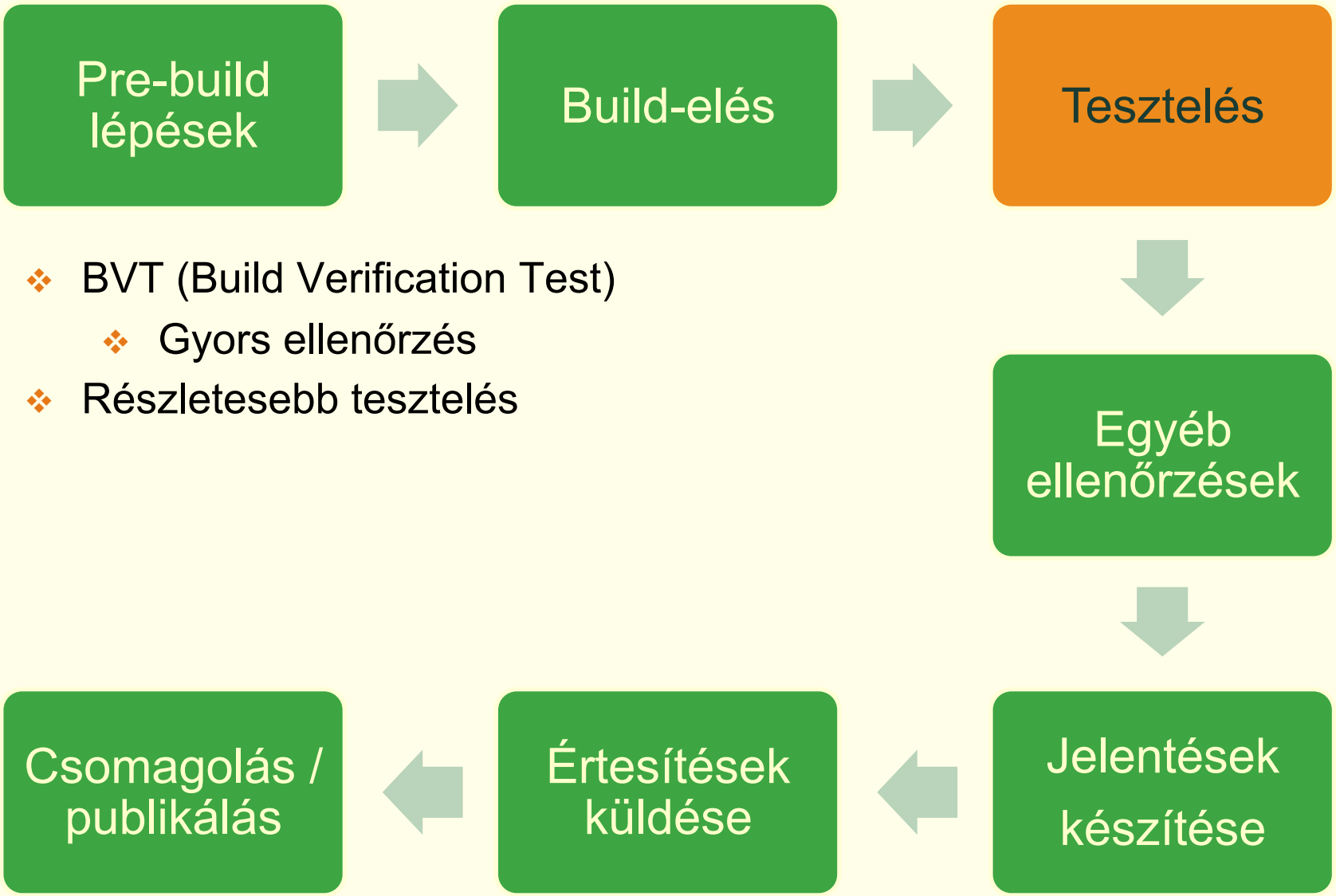
- Források beszerzése
  - SVN, CVS, fájlrendszer…
- Környezet előállítása vagy alaphelyzetbe állítása

Főbb lépések



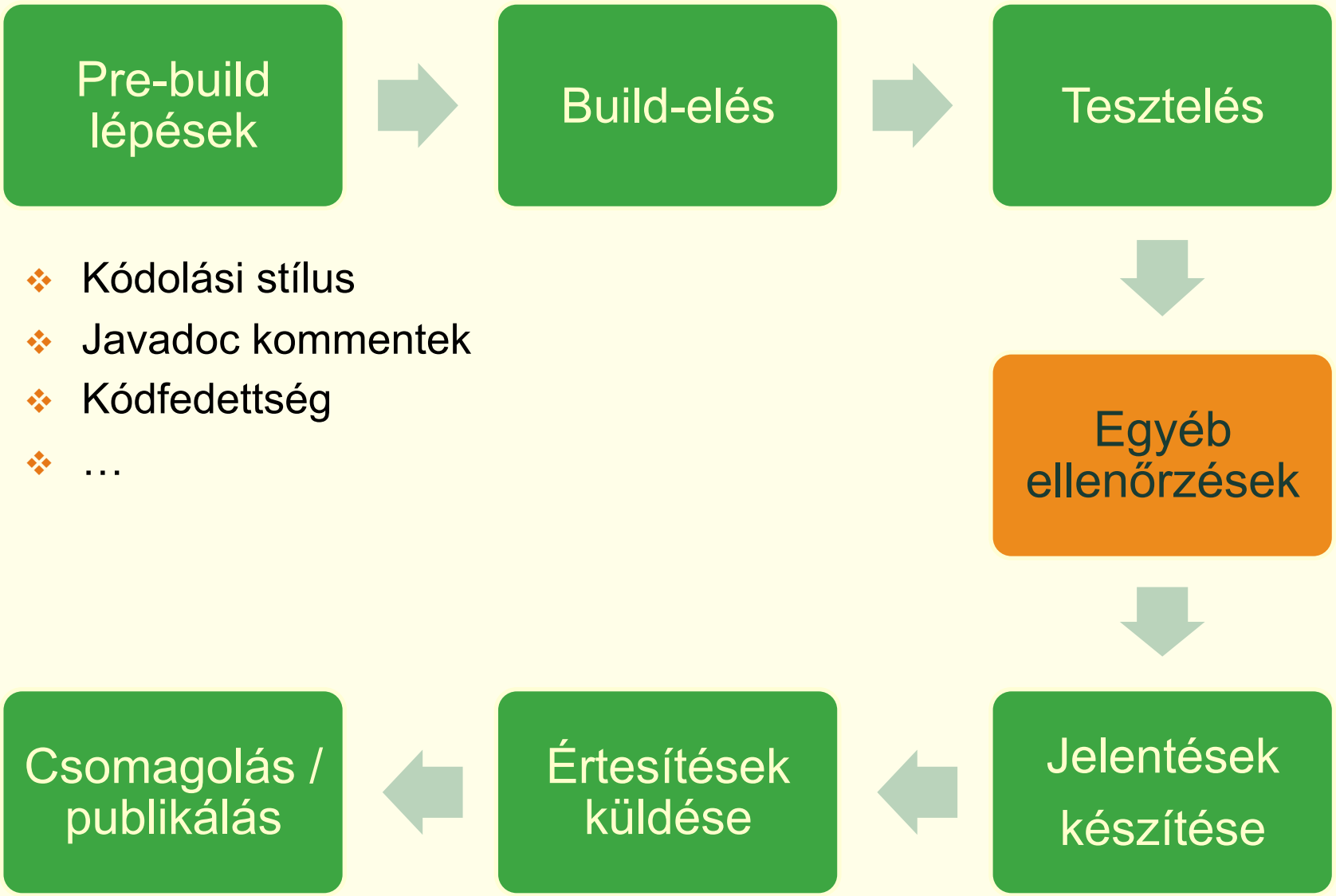
- Statikus analízis
- Függőségek kezelése
- Fordítás

Főbb lépések



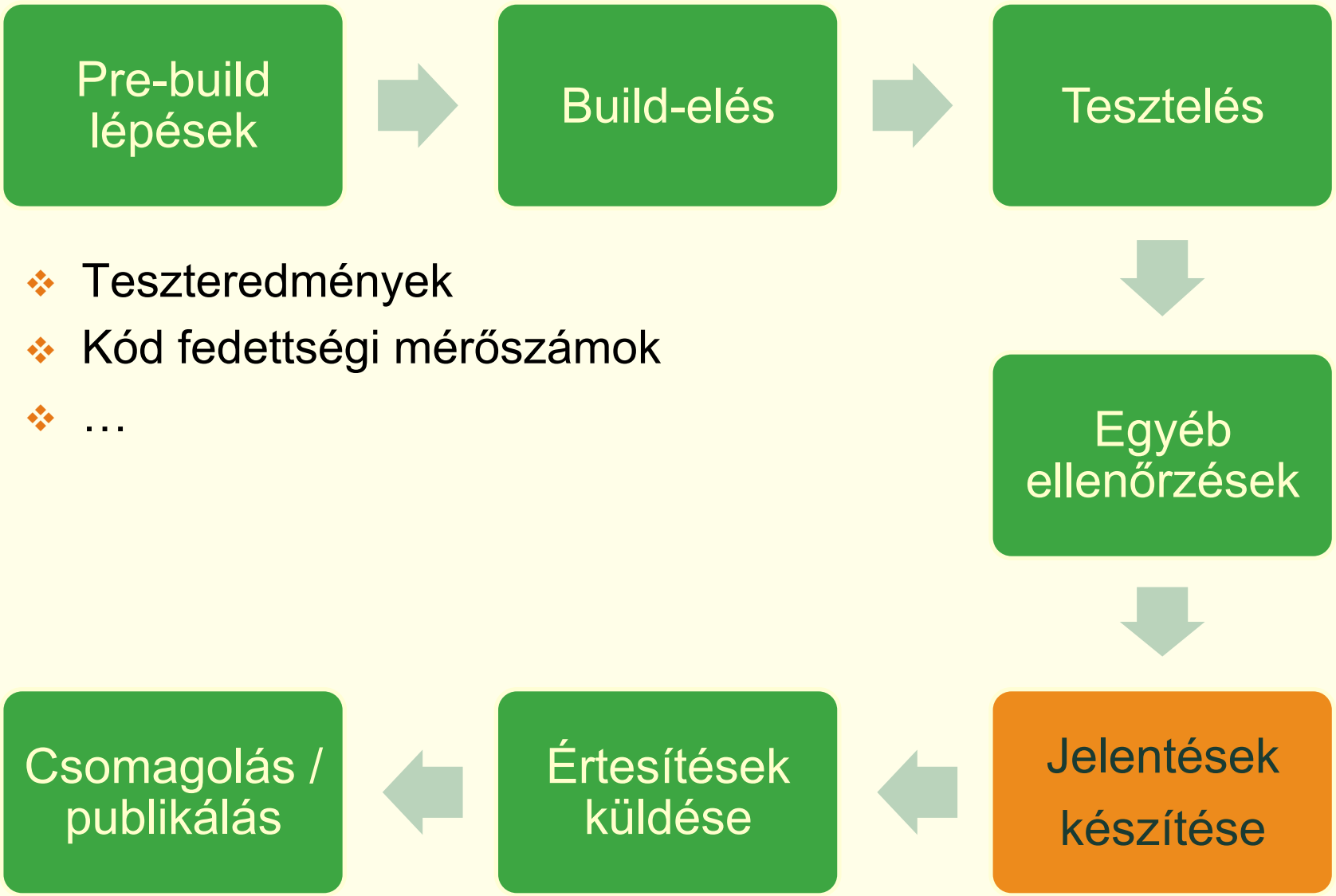
- BVT (Build Verification Test)
  - Gyors ellenőrzés
- Részletesebb tesztelés

Főbb lépések



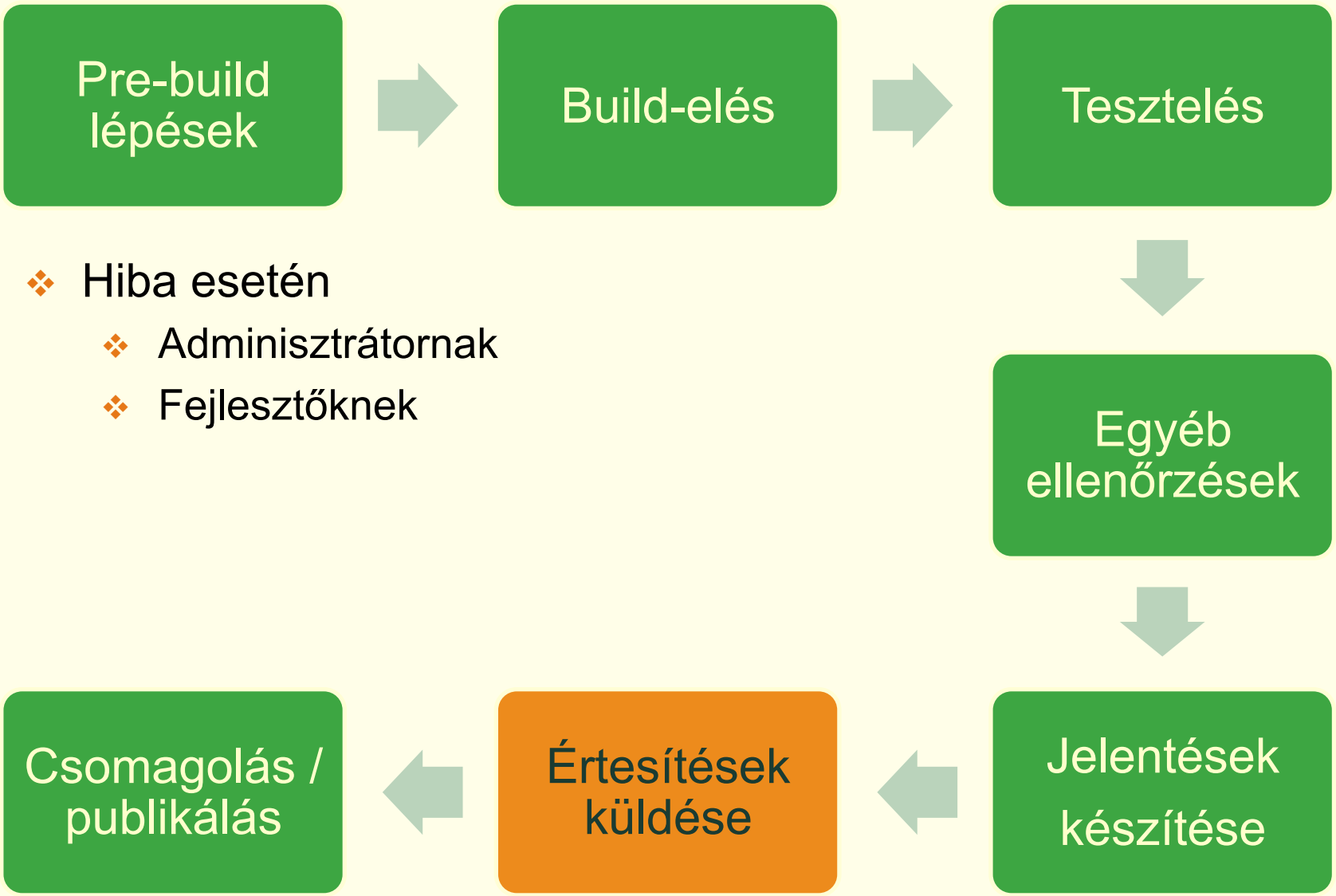
- Kódolási stílus
- Javadoc kommentek
- Kódfedettség
- …

Főbb lépések



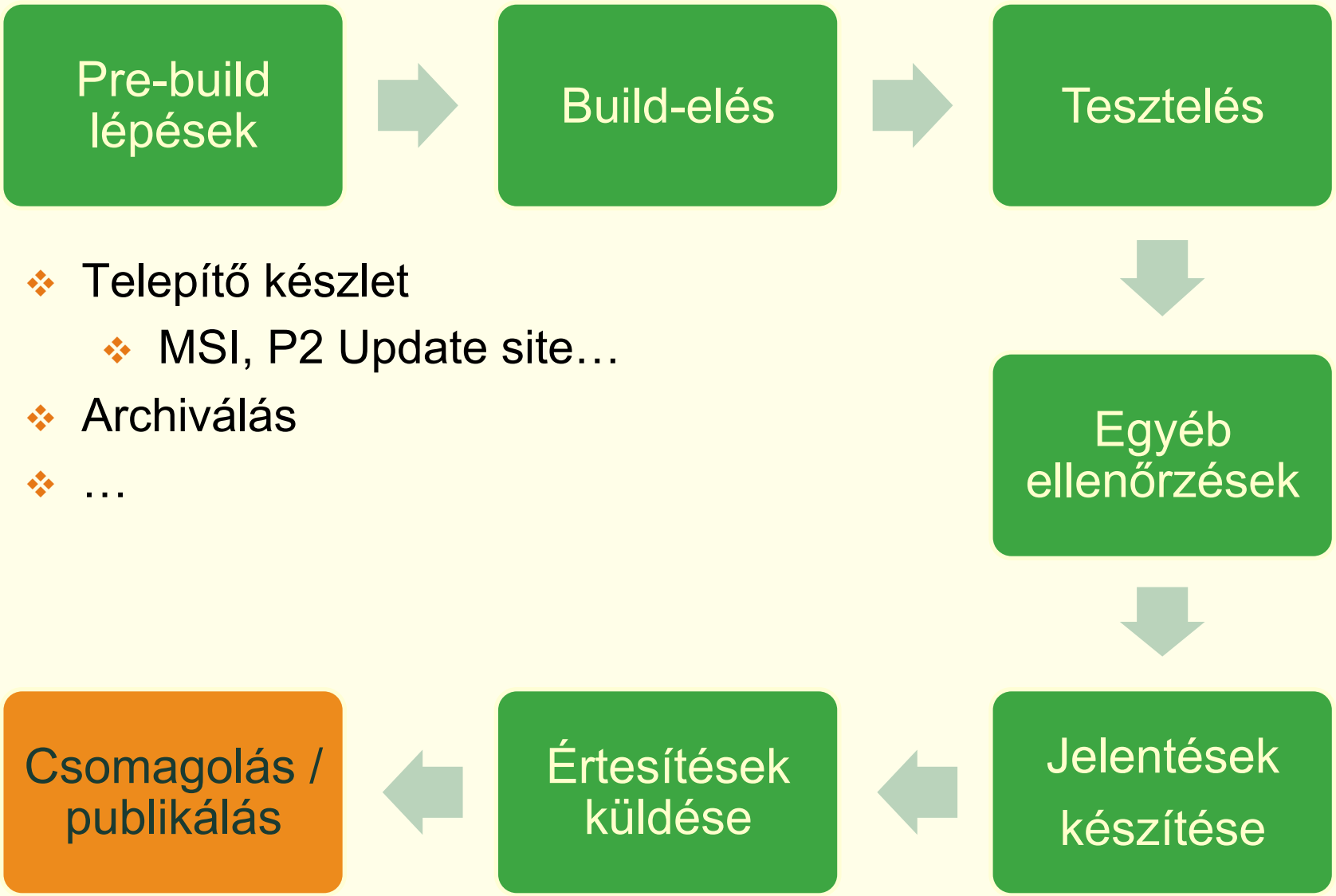
- Teszteredmények
- Kód fedettségi mérőszámok
- …

Főbb lépések



- Hiba esetén
  - Adminisztrátornak
  - Fejlesztőknek

Főbb lépések



- Telepítő készlet
  - MSI, P2 Update site…
- Archiválás
- …

Tartalom

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

Build végrehajtó motorok

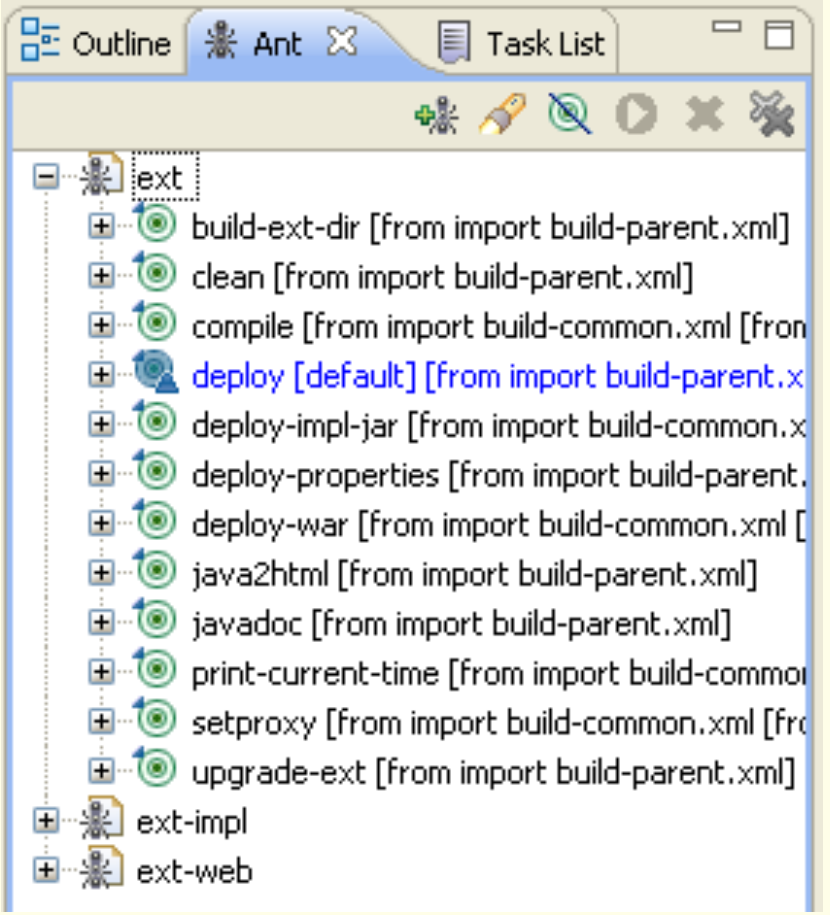
- Make
  - C/C++
- Apache Ant
  - Make fájl Java-hoz, XML alapokon
- Apache Maven
  - Egységes forrás letöltés és fordítás
  - Funkcionalitásában hasonlít az Ant-hoz
- …

Ant

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

Ant alapfogalmak

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



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

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

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>
...
```

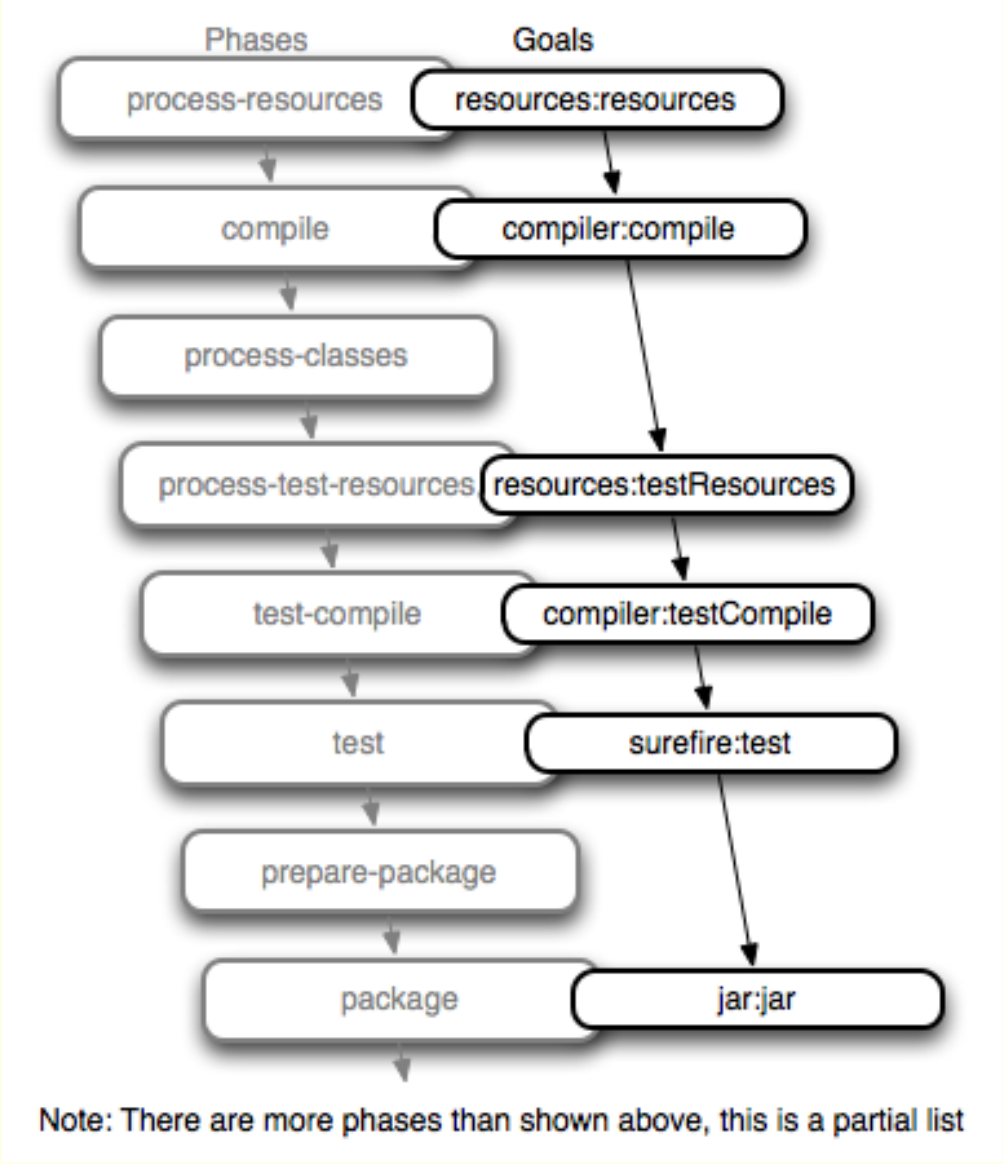
Példa (folytatás)

```
...
<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>
```

Maven

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

Maven életciklus és célok



Példa: JUnit teszt futtatás Mavennel

Projekt struktúra

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

```
<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>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Quick Start Archetype</name>
  <url>http://maven.apache.org</url>
  <packaging>jar</packaging>

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

Metaadatok

Java alkalmazás

JUnit függőség

Mi hiányzik?

Példa: JUnit teszt futtatás Mavennel

- Projekt struktúra
  - Alapértelmezett van minden alkalmazástípushoz
  - Felülbírálható
  - Ugyanakkor Maven plug-inek olvassák!

Ant vs Maven

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

Tartalom

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

Continuous Integration

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

Continous Integration

*"Continuous Integration is a software development practice where members of a team **integrate their work frequently**, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is **verified by an automated build** (including test) to detect integration errors as quickly as possible."*

Martin Fowler

http://www.martinfowler.com/articles/continuousIntegration.html

Eszközök

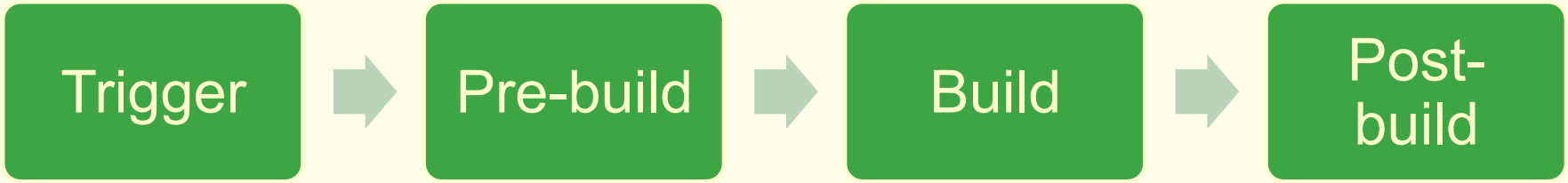
- Apache Continuum (Java)
  - XML szerkesztés + webes UI
- CruiseControl (Java, .NET, Ruby)
  - XML szerkesztés
- Hudson (Java, de kiterjeszthető)
  - Webes UI
- TeamCity (Java, .NET, Ruby)
  - Fizetős
- …

Hudson (Jenkins)

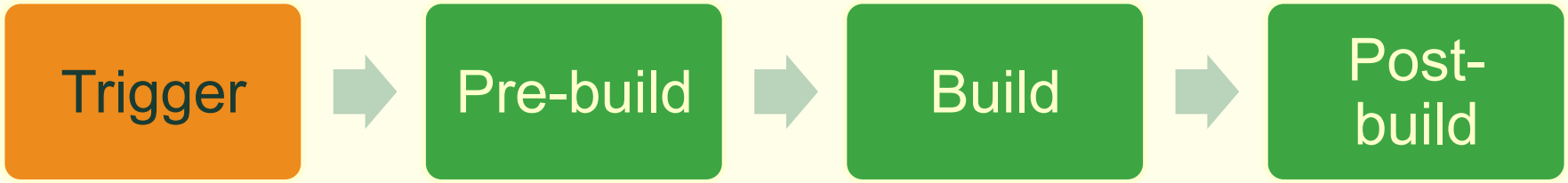
- Java szervlet alapú
  - Tetszőleges alkalmazás szerveren fut
- Plug-in alapú, **bővíthető**
- Frissítések keresése automatikus
- Gyorsan bele lehet tanulni
- Nem végez tényleges fordítást
  - Időzítés
  - Menedzselés
- Több folyamat, köztük akár **függőségekkel**

https://hudson.eclipse.org/hudson/

Hudson munkafolyamat

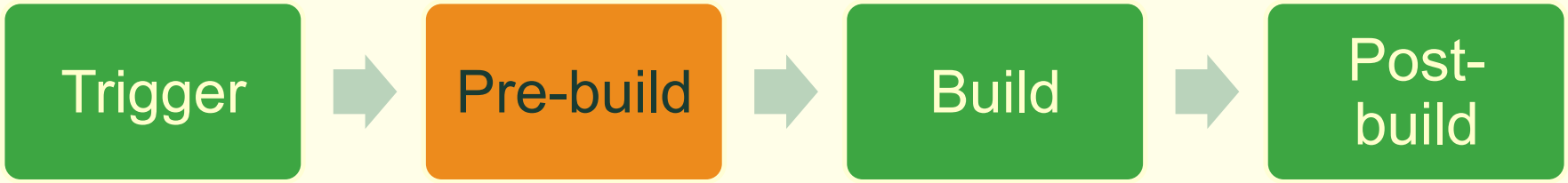


Hudson munkafolyamat



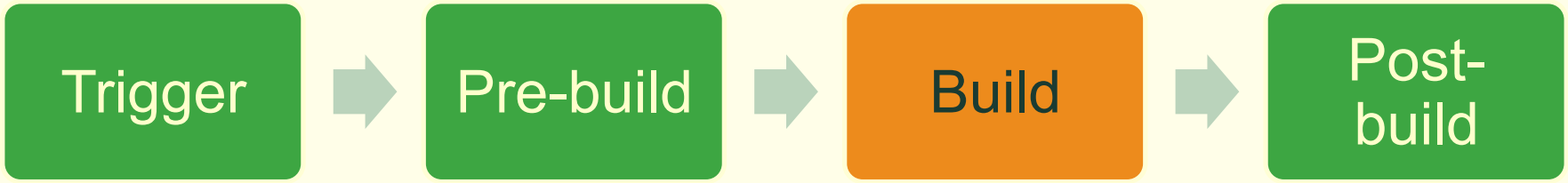
- Kézi
- Időzített
- Verziókezelő rendszer változása
- Függő job befejeződése
- Egyéb (bővíthető)

Hudson munkafolyamat



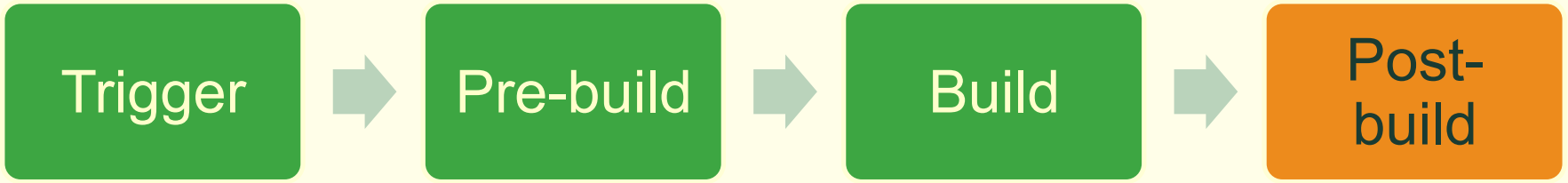
- Opcionális
- Források beszerzése

Hudson munkafolyamat



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

Hudson munkafolyamat



- Opcionális

- Archiválás
- Publikálás
- Függő build-ek indítása
- Értesítések
- …

Blame mail

Build failed in Jenkins: VIATRA2-Core #154 — Inbox

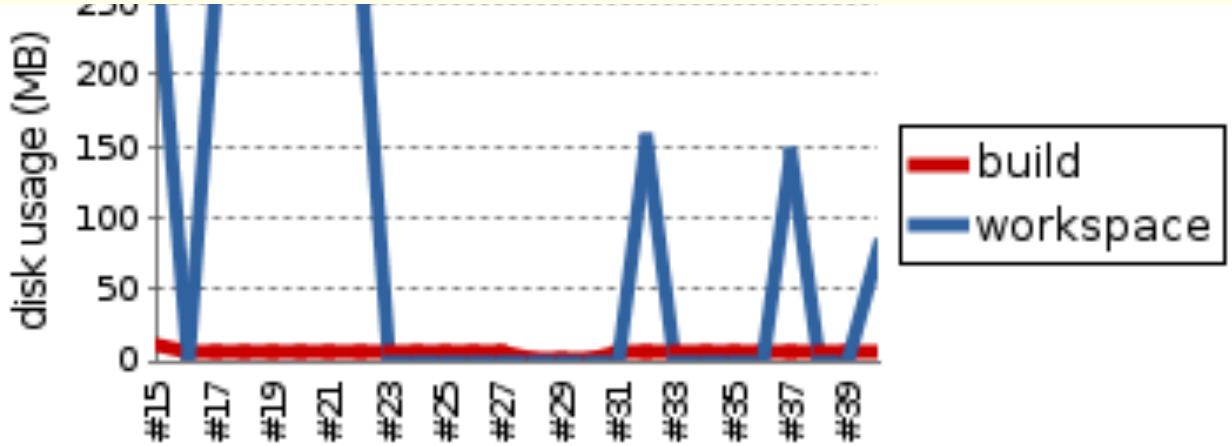
Delete Junk Reply Reply All Forward Print To Do

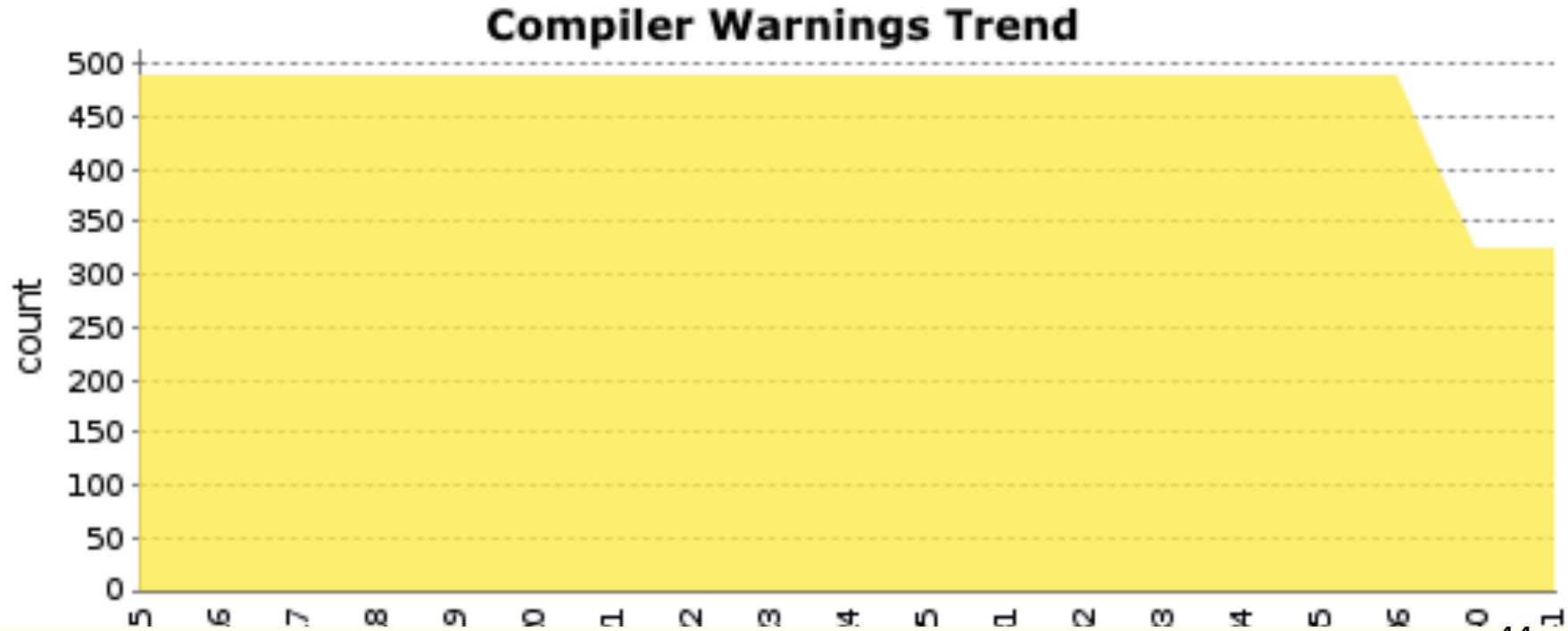
From: Jenkins Build <ujhelyiz@mit.bme.hu>
Subject: **Build failed in Jenkins: VIATRA2-Core #154**
Date: 2011. március 23. 21:34:40 CET
To: Zoltán Ujhelyi

See <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/154/>

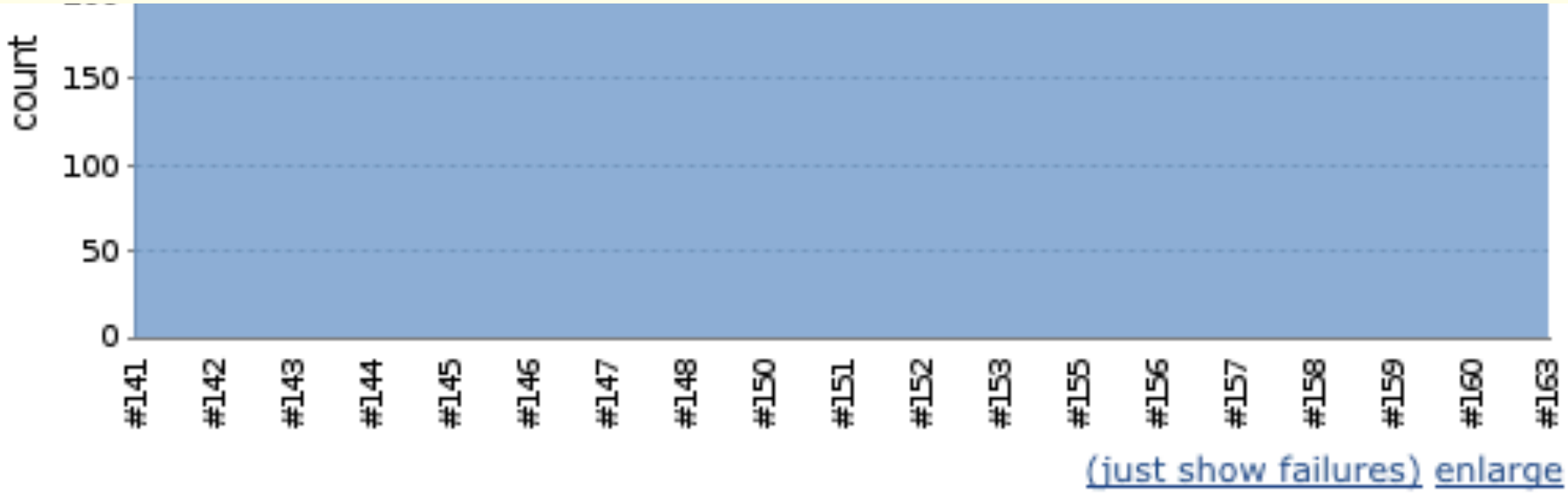
Started by user ujhelyiz
Cleaning up <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/releng>
Updating https://viatra.inf.mit.bme.hu/svn/releng/trunk
At revision 4892
Cleaning up <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core>
Deleting <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core/org.eclipse.viatra2.gtasm.interpreter.term/bin>
Deleting <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core/org.eclipse.viatra2.gtasm.patternmatcher.incremental.rete/bin>
Deleting <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core/org.eclipse.viatra2.gtasm.patternmatcher/bin>
Deleting <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core/org.eclipse.viatra2.gtasm.model/bin>
Deleting <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core/org.eclipse.viatra2.gtasm.model/src/org>
Deleting <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core/org.eclipse.viatra2.visualisation/bin>
Deleting <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core/org.eclipse.viatra2.gtasm.patternmatcher.impl/bin>
Deleting <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core/org.eclipse.viatra2.editor.text/bin>
Deleting <https://build.inf.mit.bme.hu/jenkins/job/VIATRA2-Core/ws/core/org.eclipse.viatra2.gtasm.support/bin>

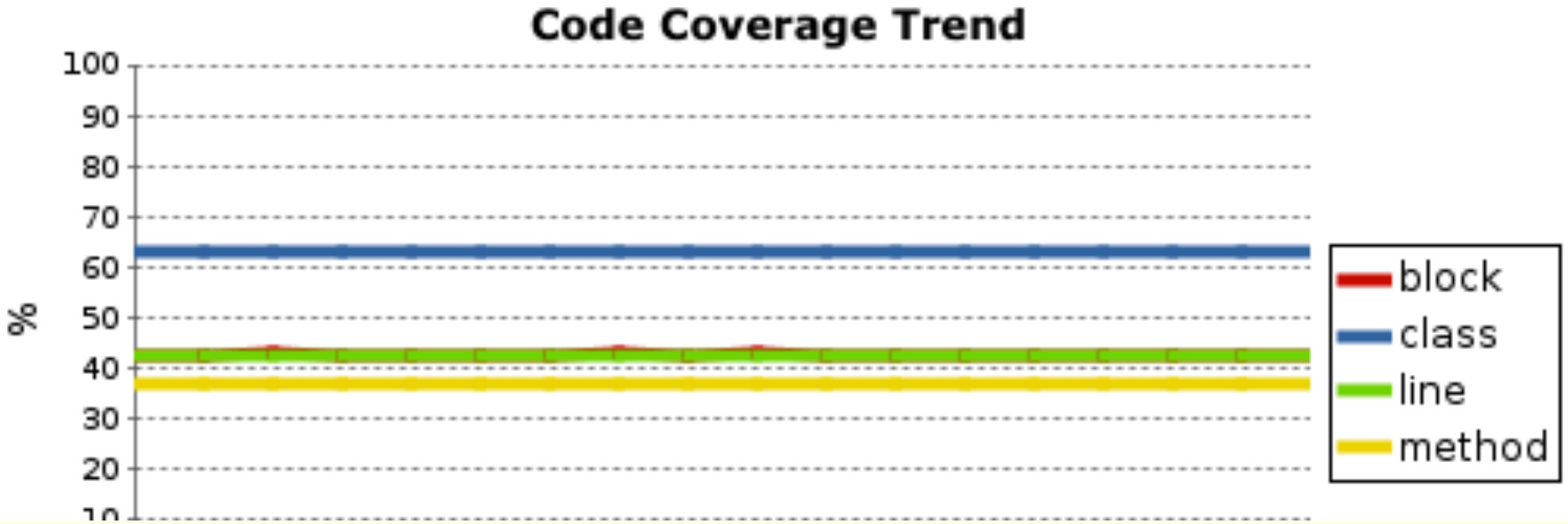
Metrikák, trendek





Kódfedettség trendek





Egyéb metrikák

Lines of code
144,398
280,278 lines
63,450 statements
2,077 files

Classes
2,199
236 packages
15,226 methods
+677 accessors

Violations
29,206

Blocker	0	
Critical	43	
Major	9,487	
Minor	15,929	
Info	3,747	

Rules compliance
69.1%

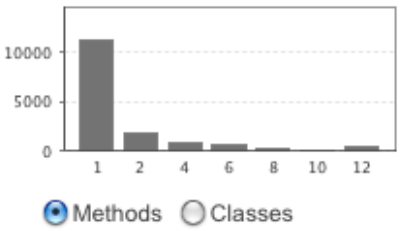
Comments
26.7%
52,600 lines
34.8% docu. API
8,554 undocu. API
3,340 commented LOCs

Duplications
10.3%
28,898 lines
12,322 blocks
673 files

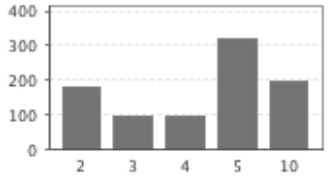
Package tangle index
22.9%
> 1,216 cycles

Dependencies to cut
186 between packages
570 between files

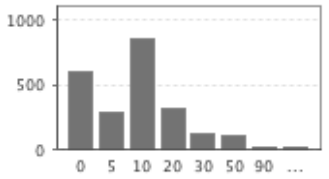
Complexity
2.4 /method
16.5 /class
17.5 /file
Total: 36,371



LCOM4
3.0 /class
41.0% files having LCOM4>1



RFC
18 /class



Code coverage
16.6%
19.1% line coverage
10.0% branch coverage
282 tests
46.7 sec

Test success
94.3%
14 failures
2 errors

További lehetőségek

- Build slave-ek
  - További Hudson példányok kezelése
- Munkafolyamatdefiníció
  - Több job egymás után
- Sokféle bővítmény
  - Trigger
  - Jelentések
  - Közzététel

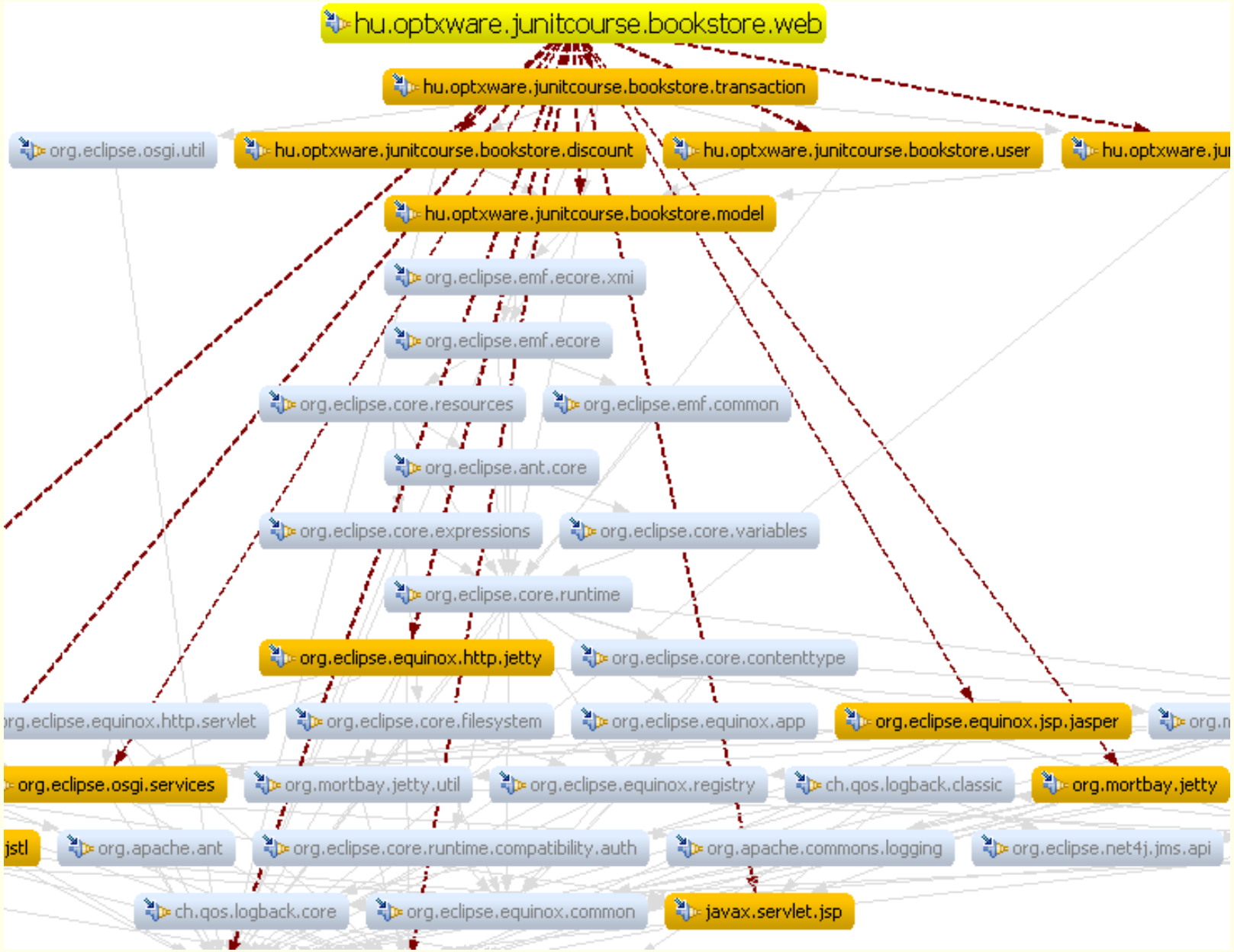
Példa: Eclipse plug-inek fordítása

- Eclipse plug-in
  - Beépülő modul
  - Jól definiált függőségek
  - Verziószámok
- De önmagában nem
  - Fordul
  - Fut
  - Tesztelésnél izoláció problémás lehet!

Példa: Eclipse plug-inek fordítása

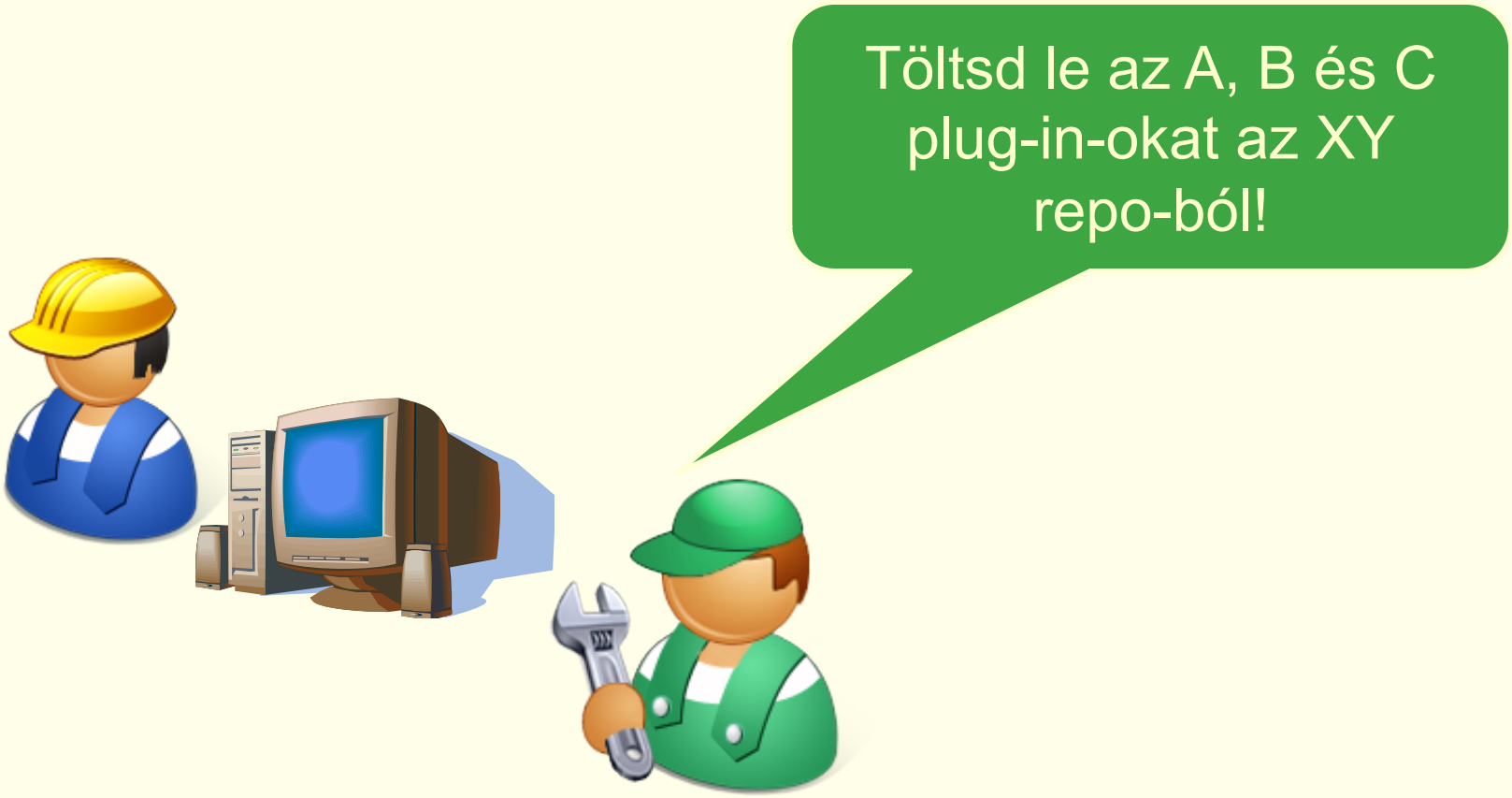
- Követelmények

Probléma


hu.optxware.junitcourse.bookstore.web
hu.optxware.junitcourse.bookstore.transaction
org.eclipse.osgi.util
hu.optxware.junitcourse.bookstore.discount
hu.optxware.junitcourse.bookstore.user
hu.optxware.ju
hu.optxware.junitcourse.bookstore.model
org.eclipse.emf.ecore.xmi
org.eclipse.emf.ecore
org.eclipse.core.resources
org.eclipse.emf.common
org.eclipse.ant.core
org.eclipse.core.expressions
org.eclipse.core.variables
org.eclipse.core.runtime
org.eclipse.equinox.http.jetty
org.eclipse.core.contenttype
org.eclipse.equinox.http.servlet
org.eclipse.core.filesystem
org.eclipse.equinox.app
org.eclipse.equinox.jsp.jasper
org.eclipse.osgi.services
org.mortbay.jetty.util
org.eclipse.equinox.registry
ch.qos.logback.classic
org.mortbay.jetty
jstl
org.apache.ant
org.eclipse.core.runtime.compatibility.auth
org.apache.commons.logging
org.eclipse.net4j.jms.api
ch.qos.logback.core
org.eclipse.equinox.common
javax.servlet.jsp

Probléma (folytatás)

Probléma (folytatás)

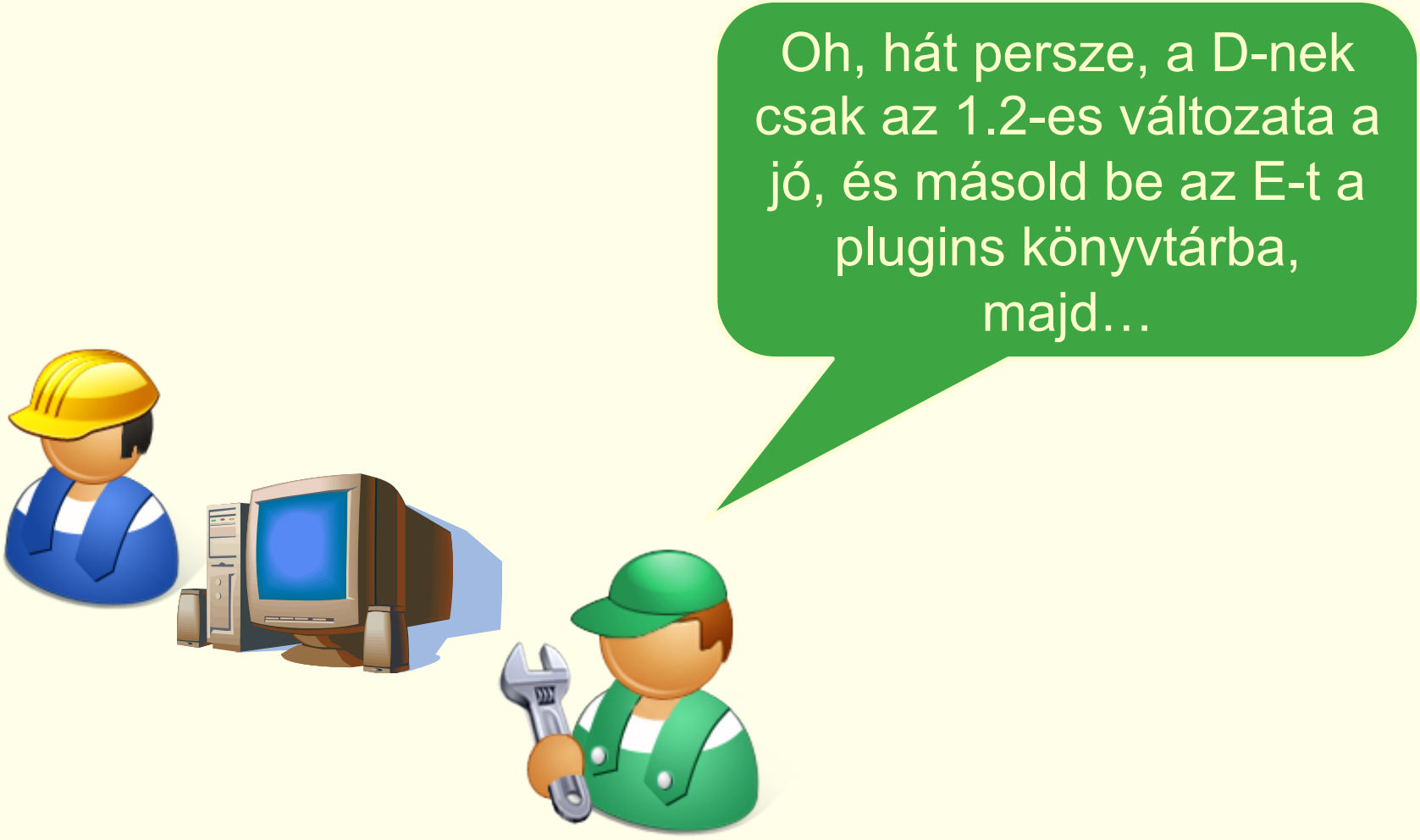


Probléma (folytatás)

Probléma (folytatás)

Probléma (folytatás)

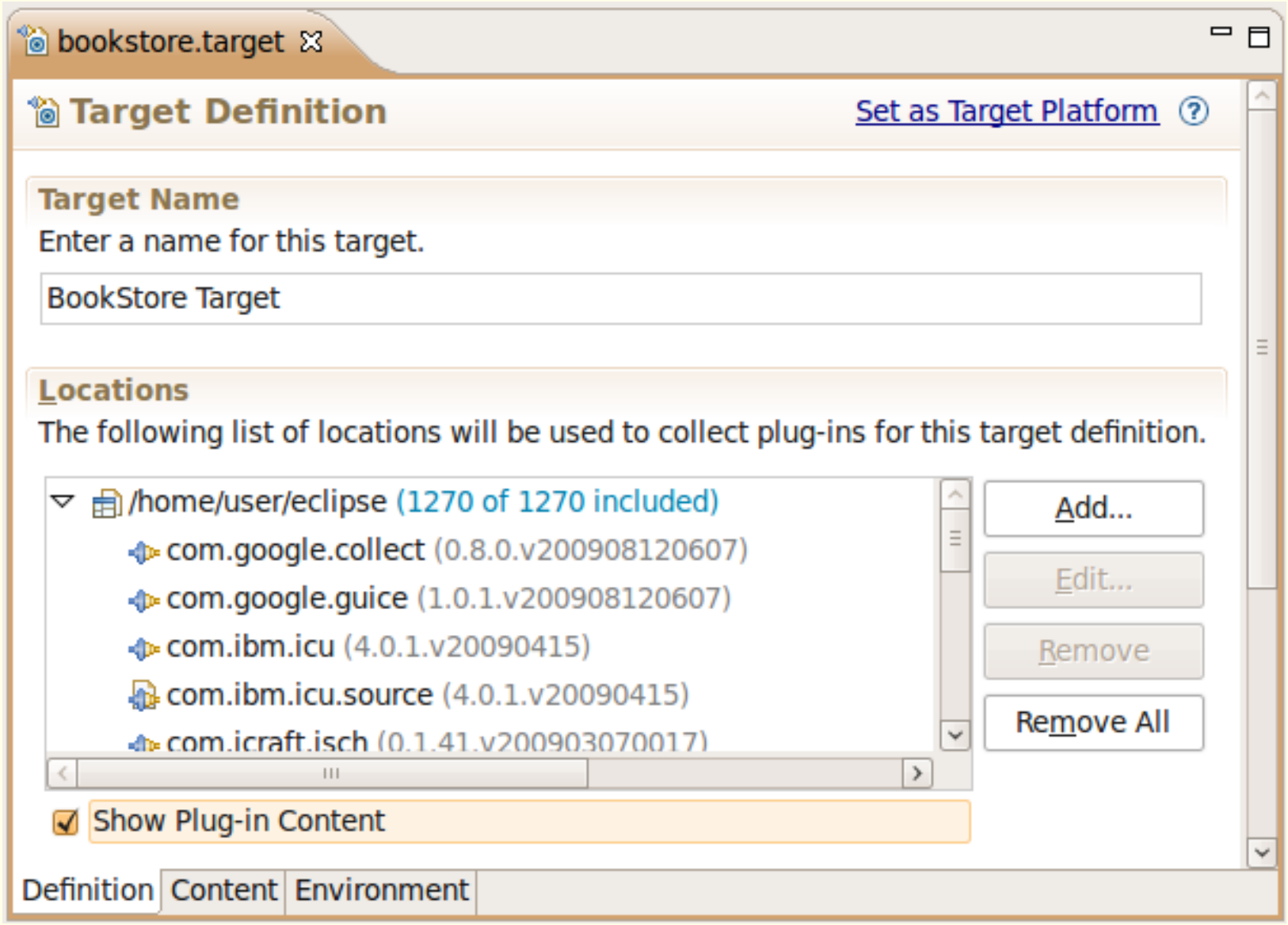
Probléma (folytatás)



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
  - …

Target Platform szerkesztő



OSGi függőségek kezelése

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

(Mindegyik bonyolult!)

Buckminster

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

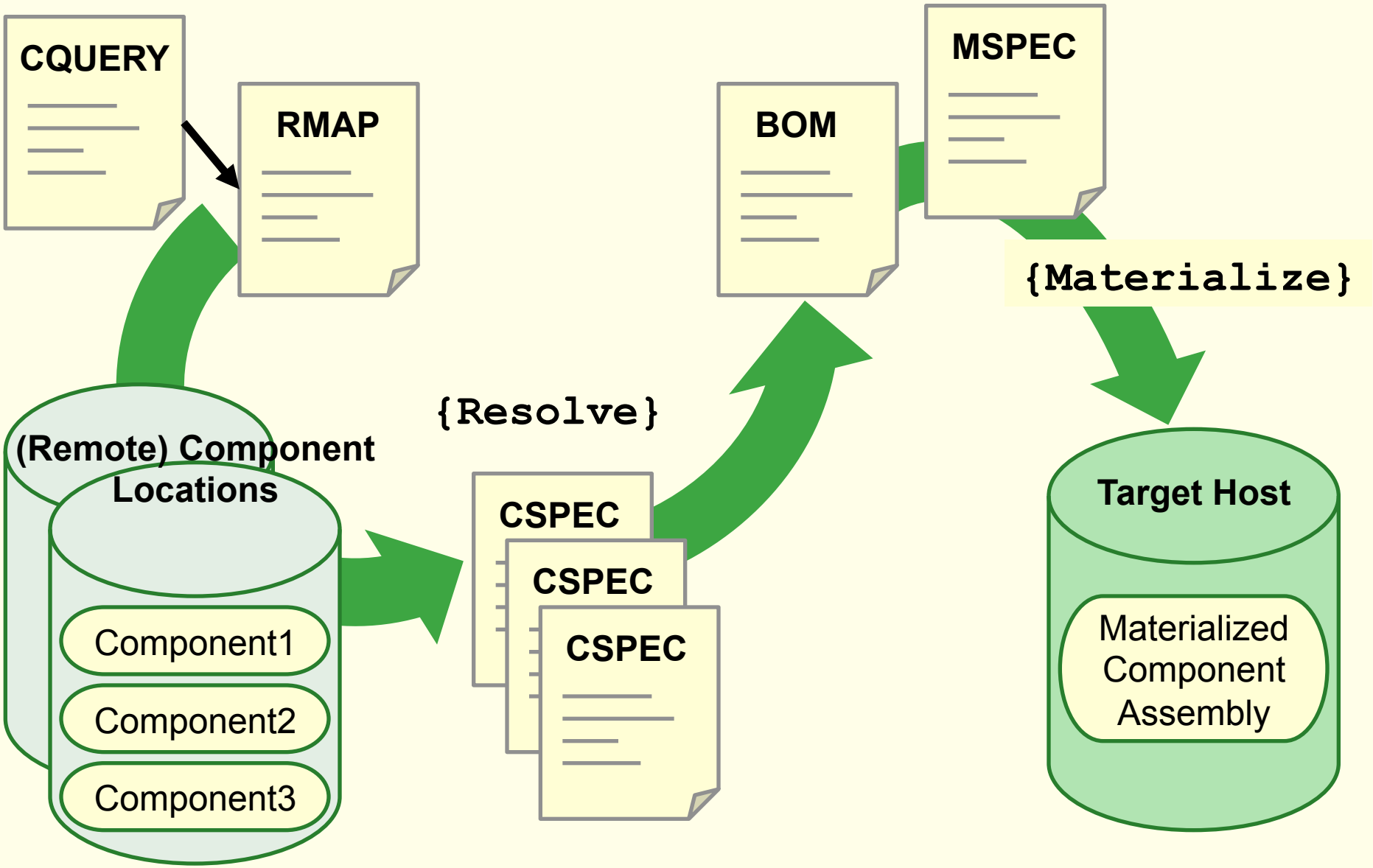
Felhasználási módok

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

Képességek

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

Leírók


CQUERY
RMAP
MSPEC
BOM
{Materialize}
(Remote) Component Locations
Component1
Component2
Component3
{Resolve}
CSPEC
CSPEC
CSPEC
Target Host
Materialized Component Assembly

Tycho

- Maven alapú build
  - Meglevő projektekből származtatható build vezérlés
- Maven csomagoló típusok
  - Plug-in
  - Plug-in teszt projekt
  - Feature
  - Repository (update site)

Elérhető példakód

- Minerva projekt
  - http://wiki.eclipse.org/Minerva
- Kipróbálás
  - Maven telepítés
  - Letöltés gitről
  - mvn clean install

Összefoglalás

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