Rendszerépítés

Jeszenszky Péter

2023.02.23.

Hivatkozások

A prezentáció az alábbi könyvön alapul:

- Ian Sommerville. *Software Engineering*. 10th ed. Pearson, 2015. https://software-engineering-book.com/
  - Chapter 25: Configuration Management (p. 730–755)

A `build` szó fordítása

- A hétköznapi fordítás az “épít”.
- Jól hangzik például a *system building* kifejezés fordításaként a rendszerépítés (szubjektív vélemény).
- Szoftverek kontextusában az “épít” fordítás kevésbé jól hangzik, lásd például *software building* (szubjektív vélemény).
- Szoftverek kontextusában az általam preferált magyar megfelelő az összeállítás.

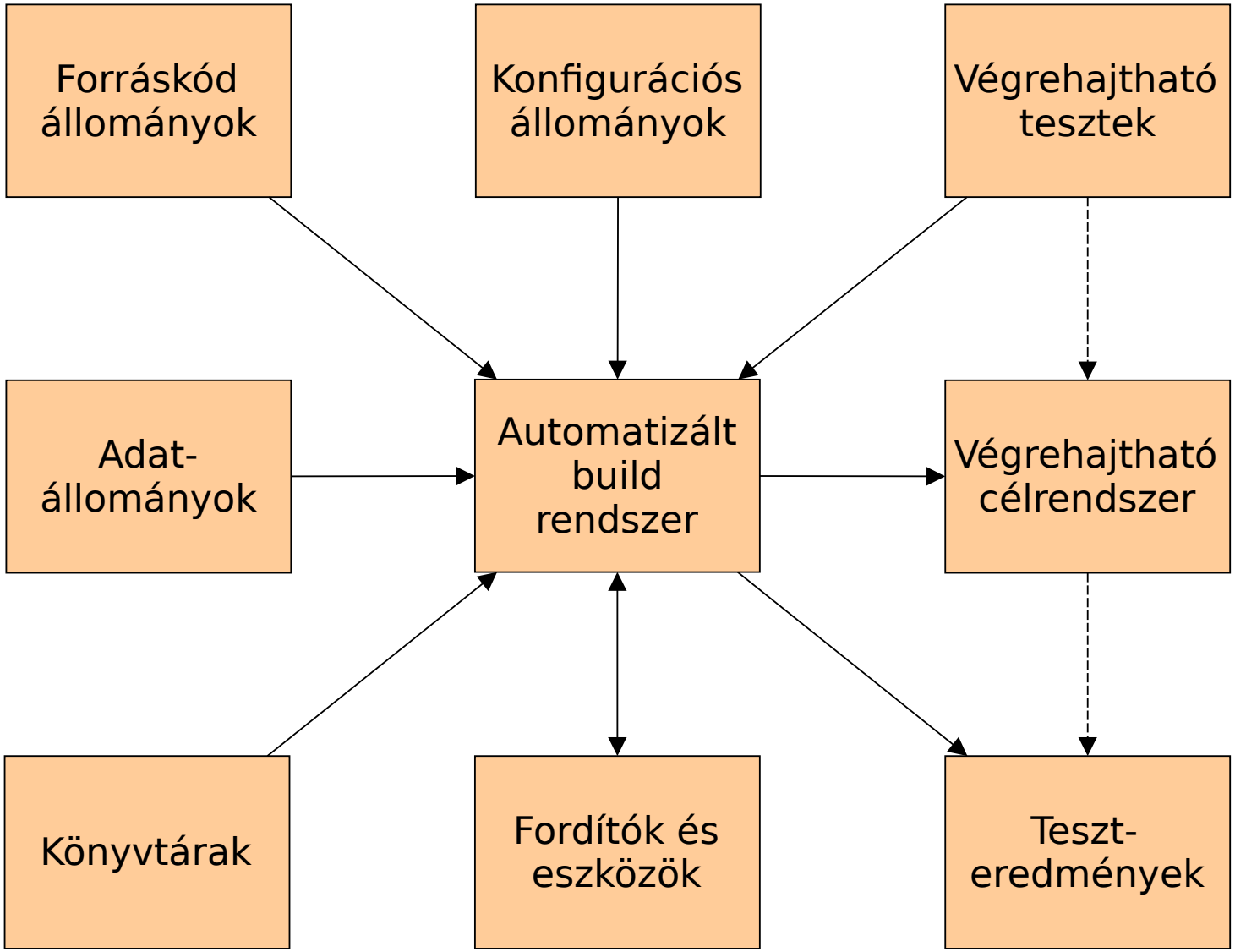
Rendszerépítés

- A rendszerépítés az a folyamat, melynek során egy teljes végrehajtható rendszer kerül létrehozásra a rendszerkomponensek, külső könyvtárak, konfigurációs állományok és további információk lefordításával és összeszerkesztésével.

Build automatizálás (1)

- A rendszerépítés nagy mennyiségű, a szoftverrel és a működési környezetével kapcsolatos információ összegyűjtésével jár. Emiatt érdemes egy automatikus *build* eszközt használni a rendszerépítéshez.
- Ideális esetben egyetlen paranccsal összeállítható egy teljes rendszer.

Build automatizálás (2)



Build eszközök funkciói (1)

- **Build szkript generálás**: A rendszernek elemeznie kell az összeállítandó programot és automatikusan generálnia kell egy *build* szkriptet (konfigurációs állományt). A rendszernek támogatni kell a *build* szkriptek kézi létrehozását és szerkesztését.
- **Verziókezelő rendszer integráció**: A rendszernek le kell tudni szedni a verziókezelő rendszerből a komponensek szükséges verzióit.
- **Minimális újrafordítás**: A rendszernek meg kell határoznia, hogy mely forráskódokat kell újrafordítani és szükség esetén el is kell végeznie a fordítást.
- **Végrehajtható rendszer létrehozása**: A rendszernek össze kell szerkesztenie a lefordított tárgykódú állományokat egymással és más szükséges állományokkal, mint például könyvtárakkal és konfigurációs állományokkal egy végrehajtható rendszer létrehozásához.

Build eszközök funkciói (2)

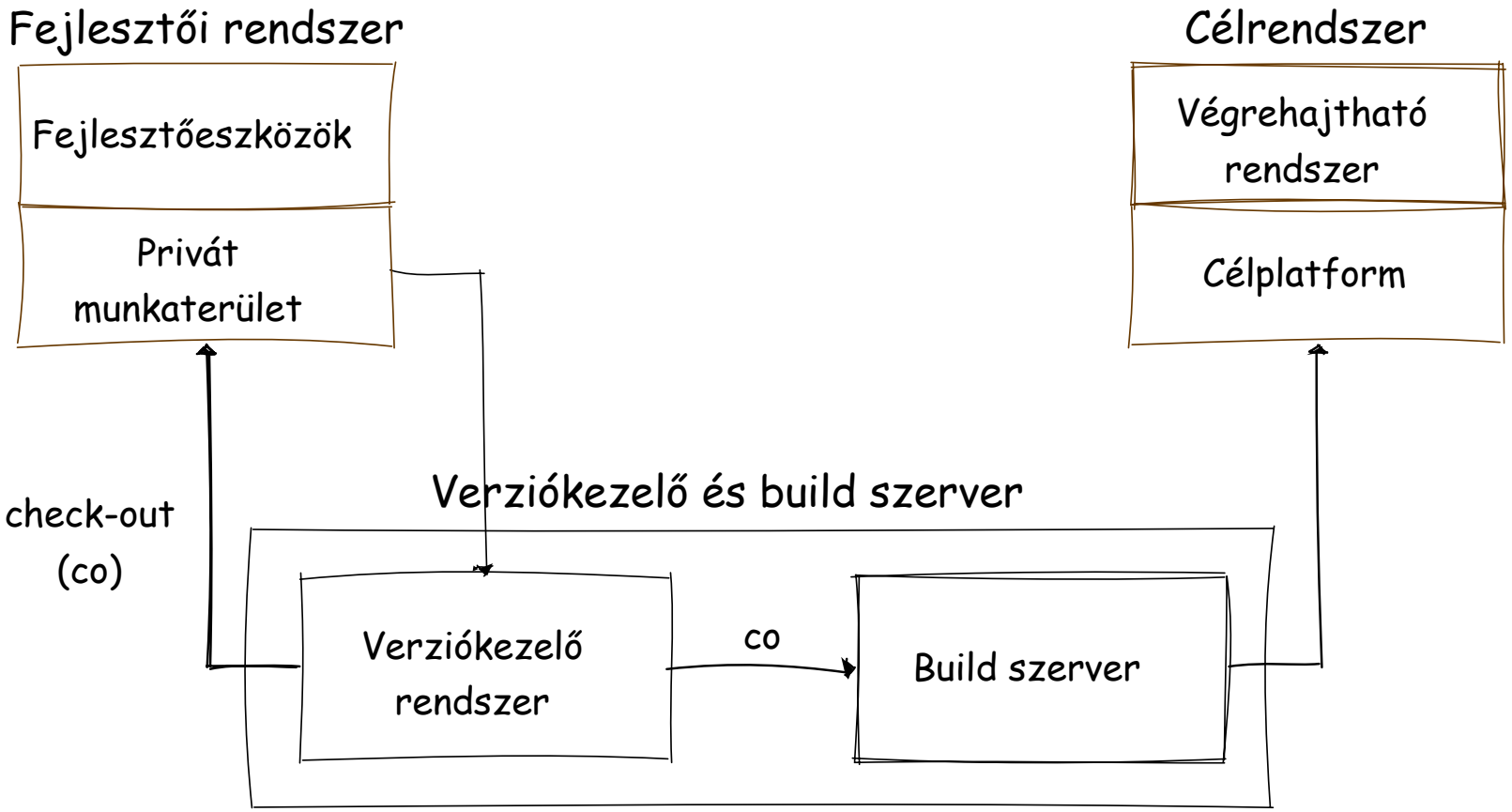
- **Teszt automatizálás**: Bizonyos rendszerek automatikusan tudnak futtatni automatikus teszteket teszt automatizálási eszközökkel, mint például a JUnit. Ezek azt ellenőrzik, hogy a változások miatt nem omlik-e össze a *build*.
- **Jelentéskészítés**: A rendszernek jelentéseket kell adnia az összeállítás sikeréről vagy sikertelenségéről és a lefuttatott tesztekről is.
- **Dokumentáció generálás**: A rendszer képes lehet különféle dokumentációk előállítására, mint például kézikönyv oldalak vagy API dokumentáció.
- **Függőségkezelés**: A rendszer képes lehet a komponensek (például könyvtárak) szükséges verzióit tárolókból letölteni.

Build szkriptek

- Egy *build* szkript az összeállítandó rendszer definíciója.
- Információkat tartalmaz a komponensekről és függőségeikről, valamint a rendszer fordításához és összeszerkesztéséhez használt eszközök verzióiról.
- Egy olyan nyelven van írva, mely konstrukciókat tartalmaz például az összeállításban részt vevő rendszerkomponensek és függőségeik leírásához.
- Példák: `Makefile` (make), `build.xml` (Apache Ant), `pom.xml` (Apache Maven), `build.gradle` (Gradle), `CMakeLists.txt` (CMake)

Az összeállítási folyamatban rész vevő platformok

Rendszerplatformok, melyek részt vehetnek az összeállítási folyamatban (Sommerville):



Folyamatos integráció (1)

Definíció (SEVOCAB):

- Módszer, mely a fejlesztett rendszer összeállításához és teszteléséhez egyetlen fővonalba fésüli össze a csapat fejlesztői által végzett forráskód módosításokat.

Folyamatos integráció (2)

Definíció (Martin Fowler):

- A folyamatos integráció egy szofverfejlesztési gyakorlat, melynél egy csapat tagjai gyakran integrálják a munkájukat. Rendszerint mindenki legalább napi rendszerességgel integrál, mely napi többszöri integrációt eredményez.
- A teszteket is magába foglaló automatikus összeállítás történik minden egyes integráció ellenőrzéséhez, hogy a lehető leghamarabb kerüljenek felfedezésre az integrációs hibák.
- Sok csapat úgy tapasztalja, hogy ez a megközelítés jelentősen csökkenti az integrációs hibákat és lehetővé teszi a csapat számára erős kohézióval bíró (*cohesive*) szoftverek gyorsabb kifejlesztését.

Folyamatos integráció (3)

- Martin Fowler. *Continuous Integration*. May 1, 2006. https://martinfowler.com/articles/continuousIntegration.html
- *Software and Systems Engineering Vocabulary (SEVOCAB)*. IEEE Computer Society, 2021. https://pascal.computer.org/

DevOps (1)

Definíció (SEVOCAB):

- Alapelvek és gyakorlatok, melyek jobb kommunikációt és együttműködést tesznek lehetővé az érdekelt felek között szoftver és rendszer termékek és szolgáltatások specifikálása, fejlesztése és működtetése céljából.

DevOps (2)

- A kifejezés a fejlesztés (*Dev*) és az üzemeltetés (*Ops*) kifejezések kombinációja.
- A DevOps a fejlesztők és az üzemeltetők összehozásáról szól a csapatokban, hogy szorosan együttműködjenek.
- A DevOps kifejezés gyakran összekapcsolódik a folyamatos integrációval (CI), a folyamatos szállítással (CD) és az infrastruktúra mint kód (Infrastructure as Code, IAC) gyakorlatával.
- A DevOps az agilis szoftverfejlesztési gyakorlatok kiterjesztése.

DevOps (3)

Ajánlott források:

- *Awesome DevOps*: http://awesome-devops.xyz/
  https://github.com/wmariuss/awesome-devops
- *DevOps Roadmap*: https://roadmap.sh/devops
  https://github.com/kamranahmedse/developer-roadmap
- Rouan Wilsenach. *DevOpsCulture*. July 9, 2015.
  https://martinfowler.com/bliki/DevOpsCulture.html
- Mikael Krief. *Learning DevOps*. 2nd ed. Packt Publishing, 2022.

Összeállítás automatizálási szoftverek listája

https://en.wikipedia.org/wiki/List_of_build_automation_software

Make (1)

- A `make` volt az első széles körben alkalmazott *build* eszköz, melyet ma is használnak.
- A `make` szabványos és kortárs megvalósítása a GNU Make.
  - Weboldal: https://www.gnu.org/software/make/
  - Programozási nyelv: C
  - Operációs rendszer: Linux, macOS
  - Licenc: GPLv3

Make (2)

- Egy olyan eszköz, mellyel végrehajtható és más nem forrás állományok hozhatók létre az alapul szolgáló forrásállományokból.
- Az eszközt egy `Makefile`-nak hívott konfigurációs állomány vezérli, mely minden egyes nem forrás állományhoz megadja a más állományokból történő előállítás módját.

Make (3)

Egy példa `Makefile`:

```
sources = $(wildcard *.md)
targets = $(sources:.md=.html)
PANDOC = pandoc
PANDOC_OPTS = -s

.PHONY: all
all: $(targets)

%.html: %.md
	$(PANDOC) $(PANDOC_OPTS) $< -o $@

.PHONY: clean
clean:
	rm -f $(targets)
```

Nevezetes build eszközök (1)

- GNU Autotools (GNU Build System) (licenc: GPL)
  - Az alábbi komponensekből áll:
    - GNU Autoconf (programozási nyelv: m4, Perl)
      https://www.gnu.org/software/autoconf/
    - GNU Automake (programozási nyelv: Perl)
      https://www.gnu.org/software/automake/
    - GNU Libtool (programozási nyelv: shell)
      https://www.gnu.org/software/libtool/
  - Konfigurációs állományok: `configure.ac`, `Makefile.am`
  - Támogatott nyelvek: C, C++
  - Vele összeállított projektek: CPython, GCC, Git, OpenJDK, VLC

Nevezetes build eszközök (2)

- CMake (programozási nyelv: C, C++; licenc: *New BSD License*)
  https://cmake.org/ https://gitlab.kitware.com/cmake/cmake
  - Konfigurációs állomány: `CMakeLists.txt`
  - Támogatott nyelvek: C, C++
  - Vele összeállított projektek: Boost C++ Libraries, KDE, MySQL
    - Lásd: https://cmake.org/success/
- Ninja (programozási nyelv: C++, Python; licenc: *Apache License 2.0*)
  https://ninja-build.org/ https://github.com/ninja-build/ninja
  - Konfigurációs állomány: `build.ninja`
  - Támogatott nyelvek: C, C++
  - Felhasználó projektek: Chromium, Google Chrome, LLVM

Nevezetes build eszközök (3)

- Apache Ant (programozási nyelv: Java; licenc: *Apache License 2.0*)
  https://ant.apache.org/ https://github.com/apache/ant
  - Konfigurációs állomány: `build.xml`
  - Támogatott nyelvek: Java
  - Vele összeállított projektek: Apache NetBeans, Apache Tomcat, Lombok
- Bazel (programozási nyelv: Java; licenc: *Apache License 2.0*)
  https://bazel.build/ https://github.com/bazelbuild/bazel
  - Konfigurációs állomány: `BUILD`/`BUILD.bazel`
  - Támogatott nyelvek: Java, C, C++, Go, Python, ...
  - Vele összeállított projektek: Angular, Selenium, TensorFlow
    - Lásd: https://bazel.build/community/users

Nevezetes build eszközök (4)

- Gradle (programozási nyelv: Groovy, Java, Kotlin; licenc: *Apache License 2.0*) https://gradle.org/ https://github.com/gradle/gradle
  - Konfigurációs állomány: `build.gradle`
  - Támogatott nyelvek: Java, Scala, Kotlin, C, C++, Groovy, ...
  - Vele összeállított projektek: Groovy, Hibernate ORM, JUnit 5, Kotlin, Spring Framework
    - Lásd: https://gradle.com/enterprise-customers/commercial/ https://gradle.com/enterprise-customers/oss-projects/
- sbt (programozási nyelv: Scala; licenc: *Apache License 2.0*) https://www.scala-sbt.org/ https://github.com/sbt/sbt
  - Konfigurációs állomány: `build.sbt`
  - Támogatott nyelvek: Scala, Java
  - Vele összeállított projektek: Play Framework, Scala

Nevezetes build eszközök (5)

- PyBuilder (programozási nyelv: Python; licenc: *Apache License 2.0*)
  https://pybuilder.io/ https://github.com/pybuilder/pybuilder/
  - Konfigurációs állomány: `build.py`
  - Támogatott nyelvek: Python
  - Vele összeállított projektek:
    https://pybuilder.io/documentation/examples
- Grunt (programozási nyelv: JavaScript; licenc: *MIT License*)
  https://gruntjs.com/ https://github.com/gruntjs/grunt
  - Konfigurációs állományok: `package.json`,
    `Gruntfile.js`/`Gruntfile.coffee`
  - Támogatott nyelvek: JavaScript
  - Vele összeállított projektek: jQuery
    - Lásd: https://gruntjs.com/who-uses-grunt

Automatizálási szerverek (1)

- Jenkins (programozási nyelv: Java; licenc: *MIT License*)
  https://www.jenkins.io/ https://github.com/jenkinsci/jenkins
  - Platform: Linux, macOS, Windows
  - Ipari felhasználások: https://www.jenkins.io/project/adopters/

Automatizálási szerverek (2)

Online szolgáltatások:

- GitHub Actions https://github.com/features/actions
  - Egy folyamatos integrációs és folyamatos szállítási (CI/CD) platform.
  - Ingyenes nyilvános tárolókhoz.
  - Dokumentáció: https://docs.github.com/en/actions