

Probléma: informatikus hiány

IT FOLYAMATOK AUTOMATIZÁLÁSA MAXIMÁLIS FOKOZATBAN



Telvice AK 4.0 – Bakai Balázs

Telvice AK 4.0 - előzmények

- **2012 óta foglalkozunk IT folyamatokat támogató automatizációs tevékenységgel (fejlesztés, tesztelés)**
 - A Dynatrace projektek kiegészítése miatt kezdtünk el ezzel foglalkozni
 - Nagyobb projektjeink: Raiffeisen Bank, KH Bank, Generali biztosító, KEKKH, KSH
 - A bevezetés jelenleg is folyamatban van több vállalatnál
 - HP Enterprise együttműködés körvonalazódik
- **Az automatizálási keretrendszerünk már a 4. generációját éli**
 - Implementációs tapasztalatok, know-how
 - Új módszertanok és technológiák beépítése
 - Szeparált fejlesztés és tesztelés támogató platform: eltérő célok
- **Ahol automatizálunk:**
 - Fejlesztői környezet: Build pipeline, Quality Gate
 - Teszt környezet: Runtime Quality Gate +**Dynatrace**
 - Éles környezet: Kubernetes, OpenShift +**Dynatrace**

■ 1. generáció: (2012)

- Folyamatos Integrációs (CI) keretrendszer: Jenkins és SonarQube eszközökkel
- Quality Gate-en volt a hangsúly

■ 2. generáció: (2015)

- Jenkins, SonarQube, JMeter, Selenium, Nexus és **Dynatrace** integráció is megjelent
- Jenkins Build pipeline plugin használata: A pipeline minden egyes lépése egy külön job volt

■ 3. generáció: (2017)

- Jenkins 2 pipeline: JenkinsFile tartalmazza a teljes pipeline-t
- BuildBreaker funkcionalitás, transzparens maven verziózás, FlywayDB használat
- CI/CD funkcionalitás

■ 4. generáció: (2019)

- Fejlesztői és tesztelői automatizáló keretrendszer szeparálása
- GitFlow és multi branch pipeline használata
- A keretrendszer komponensei Docker konténerekben futnak: Jenkins, SonarQube, Gitea, Nexus, HTTPD
- A teljes keretrendszer egy szkript segítségével telepíthető lokális/cloud; egy/több szerveres környezetbe is
- A pipeline végén egy Docker image került eltárolásra a Nexusban

Automatizált fejlesztői környezet

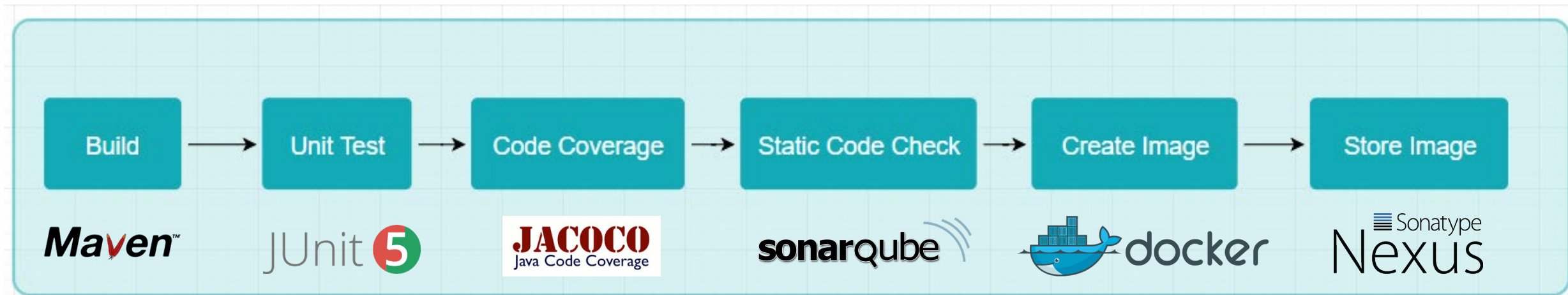
- A környezet komponensei Docker konténerekben futnak
 - Open-source, ingyenes eszközök
 - Szkripttel hozzuk létre a környezetet: lokális/cloud; egy/több szerveres kialakítás



Gitea - Git with a cup of tea



Jenkins



- Egy fejlesztési projekt a következő elemeket tartalmazza:
 - Forráskód, statikus erőforrások
 - FlywayDB scriptek: az adatbázis séma és a kód szinkronban tartásához
 - Jenkinsfile: a pipeline folyamatának (lépés sor) leírása (build → ... → store image)
 - Maven pom.xml: forráskódból a telepíthető alkalmazás létrehozása (build)
 - Dockerfile: az alkalmazás futtatásához image létrehozása

```
Dockerfile ✕  
1 FROM tomcat:8.5-alpine  
2 RUN rm -rf /usr/local/tomcat/webapps/*  
3 COPY ./target/fleetman.war /usr/local/tomcat/webapps/ROOT.war  
4 ENV JAVA_OPTS="-Dspring.profiles.active=docker-demo"  
5 EXPOSE 8080  
6 CMD ["catalina.sh", "run"]
```

A konténer alapú megközelítés előnyei

- **Nem kell többet telepítési leírást készíteni**
 - A pipeline által létrehozott Docker image-et használhatjuk minden környezetben: DEV, TEST, PROD
- **Deployment folyamatok hatékonyabbá tétele**
 - Automatizálás, folyamatok felgyorsítása, standardizálás
- **Identikus és konzisztens környezetek felállítása szkriptek segítségével**
 - „Pedig az én gépemen működött” problémák ellen
 - Fejlesztői környezet előkészítése gyors és egyértelmű egy újonnan belépő kolléga számára is
 - Az éles és a teszt ill. fejlesztői környezetek egyezése mindkét irányban biztosítható
- **Éles környezetben: Kubernetes, Openshift**
 - Auto scaling
 - Zero Down Time Deployment
 - Load balancing
 - Routing
- **Agilis, DevOps megközelítés támogatása**



Automatizált fejlesztői környezet előnyei

- **A fejlesztés korai fázisában kiderülnek a problémák:**
 - A kódok folyamatosan integrálásra kerülnek
 - Minden módosításnak át kell mennie a Quality Gate-en
 - Build Breaker funkcionalitás
- **A pipeline lépései gyorsan lefutnak**
 - Itt nem futtatunk integrációs és performancia teszteket, csak Unit teszteket
 - Visszajelzést kapunk 1-2 percen belül
- **Fokozatos átállás lehetősége:**
 - Devops alapú agilis fejlesztéshez
 - monolitikus → microservice alkalmazások fejlesztésére
 - Cloud alapú környezetre



- **A keretrendszer kialakítása a Telvice kft. know-how-ja**
- **A partnereinknél a fejlesztési-tesztelési folyamatoknál hasznosítjuk, egyedileg testre szabva**
- **Megoldásunk támogatja a dokumentáció készítésének az integritását és automatizálását**
- **Segítünk végig vinni az automatizációt:**
 - Fejlesztői környezettől kezdve
 - A teszt környezeten át
 - Az éles környezetben való üzemeltetésig



www.dynatrace.com