

ORACLE®



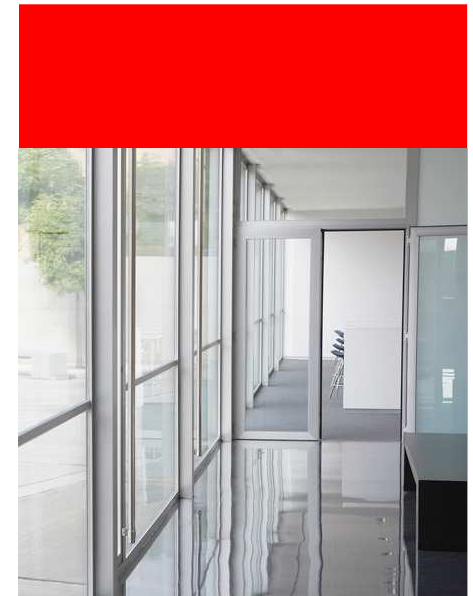
ORACLE®

Sistem za zgodnjo integracijo

Roman Naglič
Principal Consultant

Agenda

- Problemi pri razvoju programske opreme
- Rešitev
- Naloga sistema za zgodnjo integracijo
- Kako začeti
- Elementi sistema za zgodnjo integracijo
- Pravila oz. dobre prakse
- Primer sistema za zgodnjo integracijo
- Zgodnja integracija na MJU



Agenda

- Problemi pri razvoju programske opreme
- Rešitev
- Naloga sistema za zgodnjo integracijo
- Kako začeti
- Elementi sistema za zgodnjo integracijo
- Pravila oz. dobre prakse
- Primer sistema za zgodnjo integracijo
- Zgodnja integracija na MJU



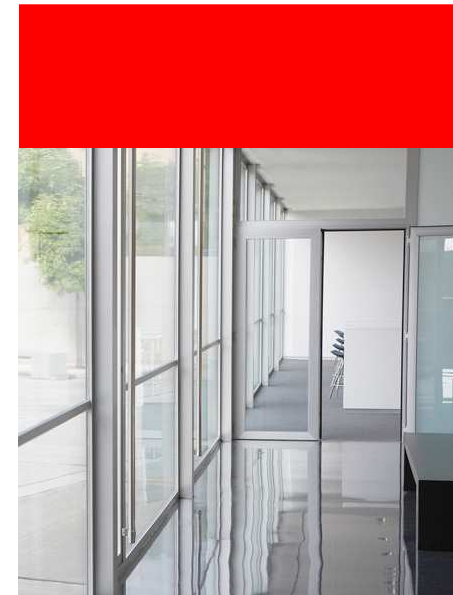


Problemi pri razvoju programske opreme

- Pozno odkrite napake v programski kodi
- Moduli med seboj niso kompatibilni
- Slaba kvaliteta programske kode
- Ročna priprava aplikacije za deployment
- Deployment na produkcijsko okolje šele po koncu razvoja
- Ročna nastavitvev aplikacije v produkcijskem okolju
- Podatkovna zbirka se namešča ročno
- Dokumentacija se ne generira avtomatsko
- Slab nadzor nad stanjem projekta (project management)

Agenda

- Problemi pri razvoju programske opreme
- **Rešitev**
- Naloge sistema za zgodnjo integracijo
- Kako začeti
- Elementi sistema za zgodnjo integracijo
- Pravila oz. dobre prakse
- Primer sistema za zgodnjo integracijo
- Zgodnja integracija na MJU





Rešitev

Proces zgodnje integracije ali še bolje proces zgodnjega oz. neprestanega deploymenta

- Stabilno okolje za pripravo aplikacije
- Zmanjšamo tveganja
- Napake so odkrite v zgodnji fazi razvoja
- Boljša kakovost programske kode
- Stanje kode je dokumentirano
- Boljši nadzor nad stanjem projekta
- Aplikacija je vedno pripravljena na deployment
- Integracijski testi so opravljeni
- Integracija vedno obsega majhne dele aplikacije in je zato bolj obvladljiva
- Nastavitve aplikacije potekajo avtomatsko

Agenda

- Problemi pri razvoju programske opreme
- Rešitev
- **Naloge sistema za zgodnjo integracijo**
- Kako začeti
- Elementi sistema za zgodnjo integracijo
- Pravila oz. dobre prakse
- Primer sistema za zgodnjo integracijo
- Zgodnja integracija na MJU





Naloge sistema za zgodnjo integracijo

- Prevajanje programske kode, konfiguriranje in izgradnja binarnega paketa iz različnih modulov (dependency management)
- Izvajanje avtomatskih testnih vzorcev
- Statično preverjanje kode
- Avtomatski deployment
- Izvajanje avtomatskih integracijskih testov
- Generiranje dokumentacije
- Avtomatsko kreiranje in nadgradnja podatkovne zbirke
- Povezava s sistemi za spremljanje napak

Agenda

- Problemi pri razvoju programske opreme
- Rešitev
- Naloga sistema za zgodnjo integracijo
- **Kako začeti**
- Elementi sistema za zgodnjo integracijo
- Pravila oz. dobre prakse
- Primer sistema za zgodnjo integracijo
- Zgodnja integracija na MJU



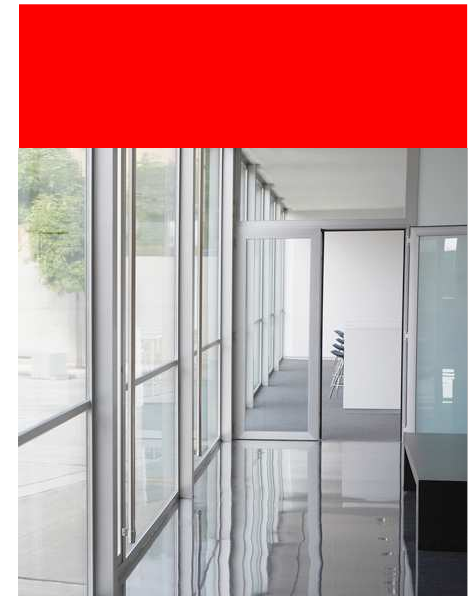


Kako začeti

- Najbolj pomembno
 - Izobraževanje in privoljenje
 - Postopnost
 - Sistem za verzioniranje
 - Continuous improvement

Agenda

- Problemi pri razvoju programske opreme
- Rešitev
- Naloge sistema za zgodnjo integracijo
- Kako začeti
- **Elementi sistema za zgodnjo integracijo**
- Pravila oz. dobre prakse
- Primer sistema za zgodnjo integracijo
- Zgodnja integracija na MJU





Elementi sistema za zgodnjo integracijo

- Sistem za hrambo in verzioniranje izvorne kode (SVN, CVS, GIT, Mercurial)
 - Vse naj gre v sistem razen binarnih datotek
 - Redni commit v glavno vejo
 - Uporabljajte smiselne commit messages
- Scripta za izgradnjo aplikacije (Ant, Maven, makefile, rake, shell scripts, ...)
 - Maven
 - Standardna direktorijska struktura
 - Obvladovanje odvisnosti
 - Lokalni in centralni repozitorij
 - Repository proxy (Artifactory, Nexus,...)



Elementi sistema za zgodnjo integracijo

- Build server (Hudson, Continuum, Cruise Control, CruiseControl.NET ...)
 - Enostavna namestitvev
 - Veliko število plugin-ov.
 - Distribuiran build sistem
 - Open source
 - Zrel produkt
- Sistem za obvladovanje odvisnosti (maven, ivy, ...)
 - Maven
 - Projekt uporablja le knjižnice, ki jih dejansko rabi
 - Seznam odvisnosti
 - Knjižnice se nahajajo na enem mestu
 - Enostavno definiranje odvisnosti





Elementi sistema za zgodnjo integracijo

- Sistem za spremljanje napak
 - Če je možno naj bo povezan s sistemom za zgodnjo integracijo
 - Centraliziran pregled števila in stanja napak sistema
 - Centralno dodeljevanje reševanja napak
 - Prioritete reševanja napak
 - BugZilla, Jira, ...
- Testno okolje, ki je čimbolj podobno produkcijskemu
 - Izvedemo lahko teste, ki preverijo konfiguracijo sistema
 - Izvedemo lahko avtomatski deployment
 - Pri vsakem buildu lahko testiramo integracijo z ostalimi sistemi

Agenda

- Problemi pri razvoju programske opreme
- Rešitev
- Naloge sistema za zgodnjo integracijo
- Kako začeti
- Elementi sistema za zgodnjo integracijo
- **Pravila oz. dobre prakse**
- Primer sistema za zgodnjo integracijo
- Zgodnja integracija na MJU



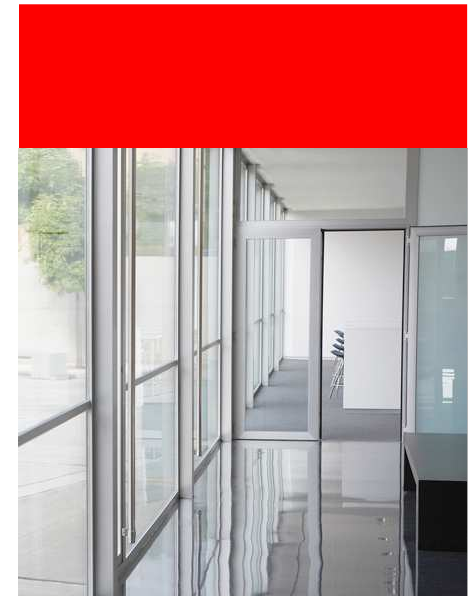


Pravila oz. dobre prakse

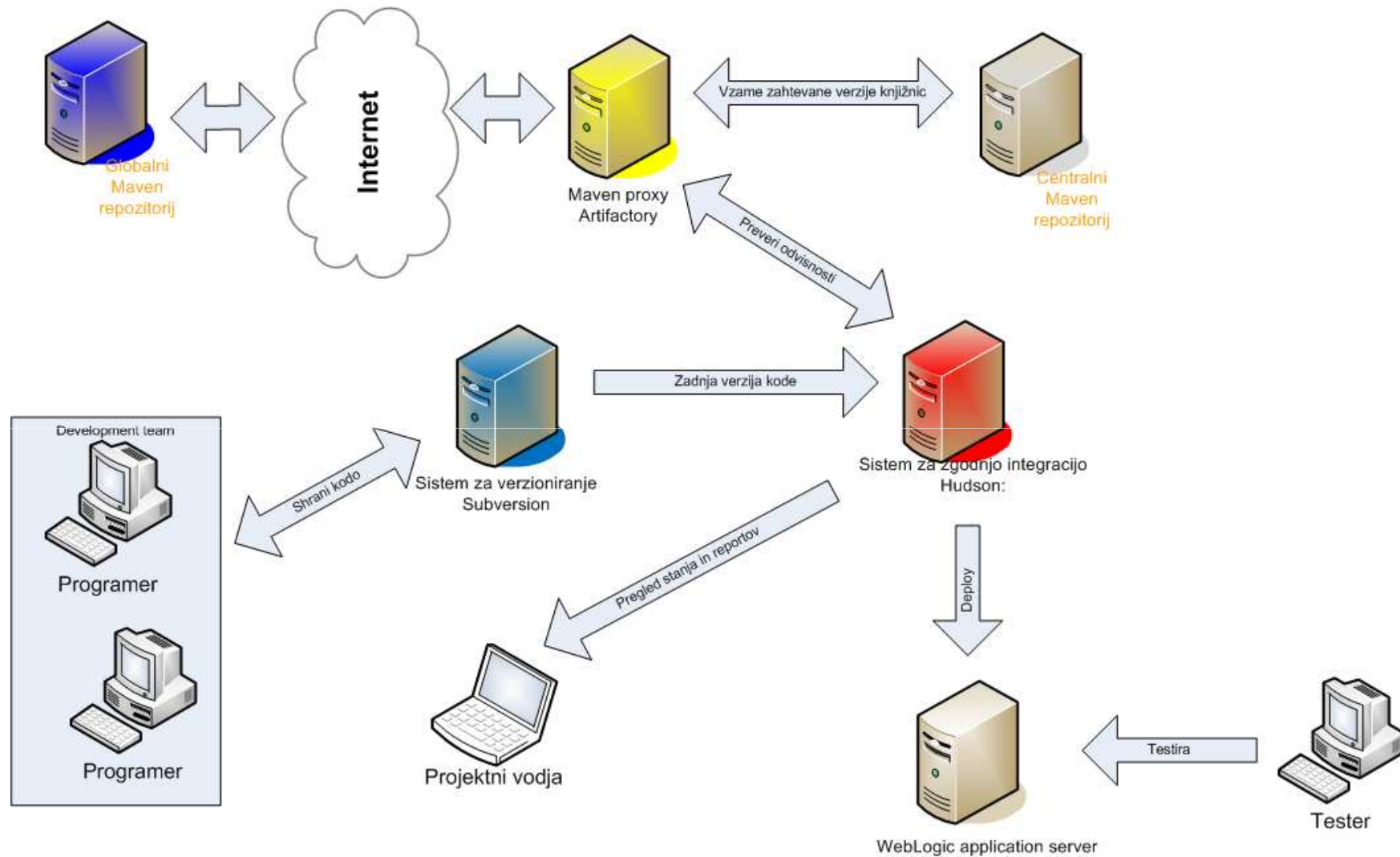
- Vsa izvorna koda ter skripte naj bodo v **enem** sistemu za verzioniranje
- Avtomatizirajte gradnjo aplikacije
- Pripravite teste za avtomatsko testiranje
- Vsak naj prenese zadnjo izvorno kodo v sistem za verzioniranje vsak dan
- Vsak prenos izvorne kode naj požene sistem za zgodnjo integracijo
- Gradnja aplikacije naj bo hitra
- Testirajte v sistemu podobnem produkcijskemu
- Dostop do zadnje verzije binarne aplikacije naj bo enostaven
- Vsak lahko vidi kaj se dogaja med gradnjo aplikacije
- Avtomatizirajte deployment

Agenda

- Problemi pri razvoju programske opreme
- Rešitev
- Naloge sistema za zgodnjo integracijo
- Kako začeti
- Elementi sistema za zgodnjo integracijo
- Pravila oz. dobre prakse
- **Primer sistema za zgodnjo integracijo**
- Zgodnja integracija na MJU



Primer sistema za zgodnjo integracijo (Hudson)

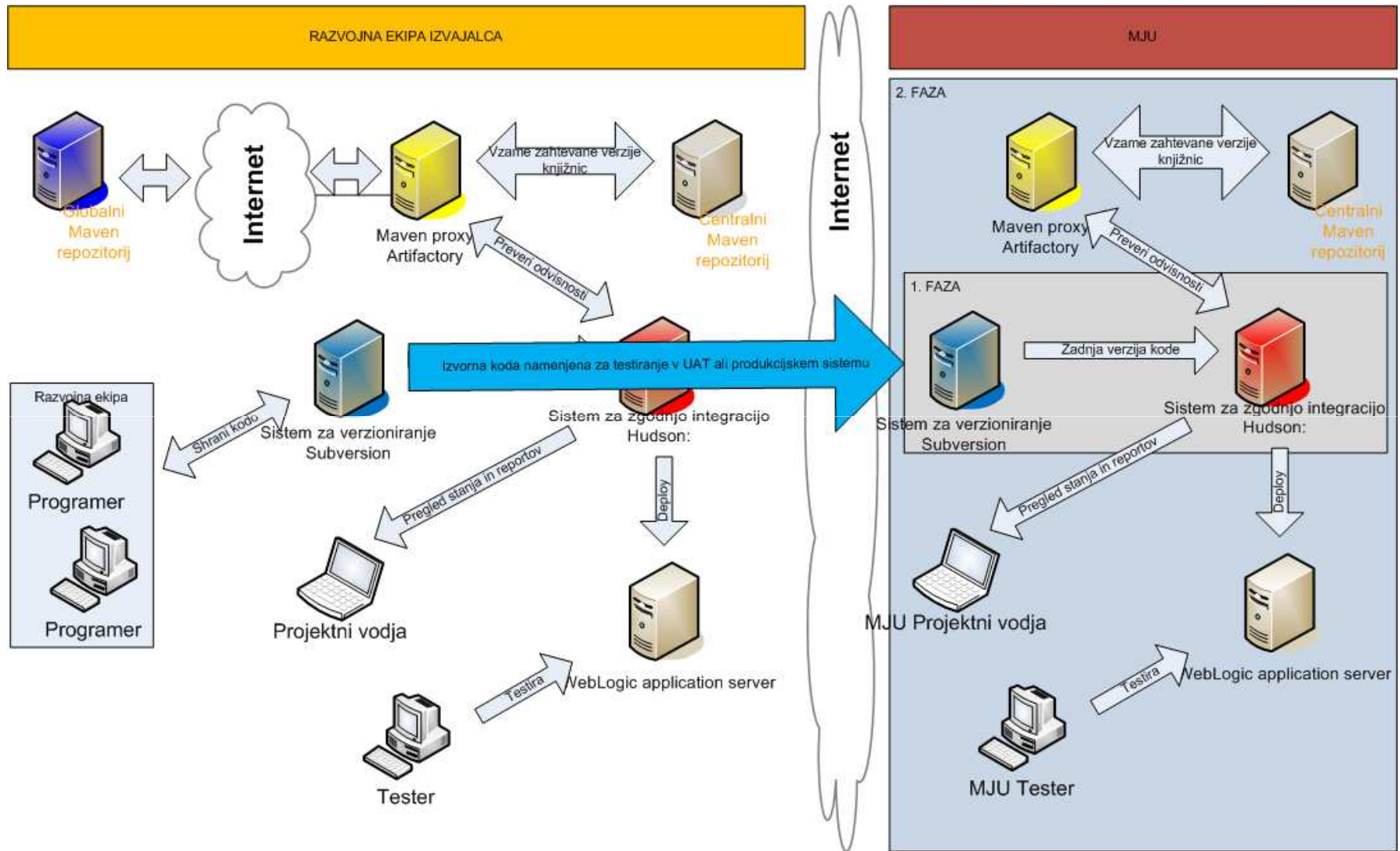


Agenda

- Problemi pri razvoju programske opreme
- Rešitev
- Naloga sistema za zgodnjo integracijo
- Kako začeti
- Elementi sistema za zgodnjo integracijo
- Pravila oz. dobre prakse
- Primer sistema za zgodnjo integracijo
- **Zgodnja integracija na MJU**



Zgodnja integracija na MJU



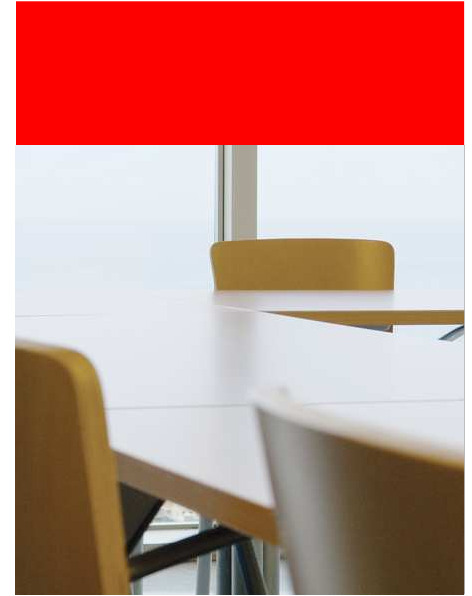


Zgodnja integracija na MJU

- V 1. fazi samo prenos izvorne kode in gradnja aplikacije
- V 2. fazi
 - analiza izvorne kode
 - centralni repozitorij
 - povezava s sistemi za spremljanje napak
 - Avtomatska namestitev aplikacije
- Prednosti za izvajalce
 - Ni več potrebe pošiljanja binarnih datotek
 - Aplikacija se zgradi v okolju enakem produkcijskemu
 - Integracija aplikacije se izvaja zgodaj, zato prej odkrite napake
 - Avtomatski testi se izvajajo v skoraj produkcijskem okolju
- Prednosti za MJU
 - Dostop do katerekoli verzije aplikacije je enostaven
 - Enostavna namestitev aplikacije
 - Testiranje kvalitete izvorne kode se izvaja pri vsaki gradnji aplikacije



Q&A



ORACLE®