

# Generičko testiranje ADF poslovnih komponenti

Jovan Stevanović, IN2

## Agenda

- Uvod
- Proces testiranja
- Oracle ADF
- Testiranje ADF poslovnih komponenti
- Generičko testiranje
- Testiranje u funkciji kontinualne integracije
- Primena u praksi
- Diskusija

## IN2 u brojkama

- 23 godine iskustva
- 12 povezanih preduzeća
- 5 zemalja
- 500+ zaposlenih
- 400+ sertifikata
- 300+ projekata
- 200+ klijenata



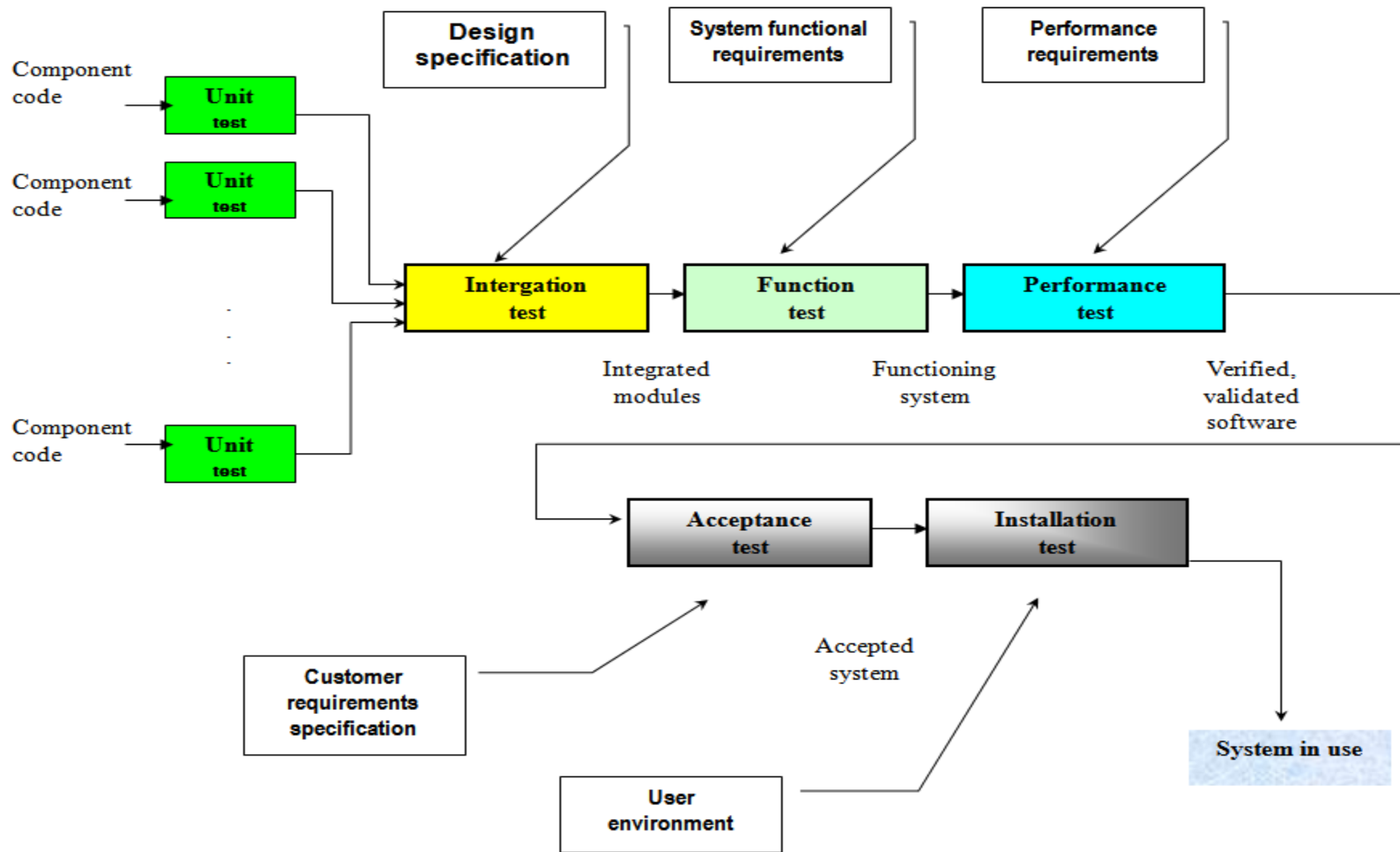
## Proces testiranja

- Evolucija pojma testiranje (po **Boris Beizer-u**)
  - FAZA - 0. Ne postoji razlika između testiranja i debugiranja, bez debugiranja testiranje nema smisla.
  - FAZA - 1. Cilj testiranja je pokazati da softver radi.
  - FAZA - 2. Cilj testiranja je **POKAZATI DA SOFTVER NE RADI!**
  - FAZA - 3. Cilj testiranja nije da dokaže **BILO ŠTA** već da smanji **RIZIK DA SOFTVER NE RADI** na prihvatljiv nivo.
  - FAZA - 4. Testiranje nije čin. To je mentalna disciplina koja rezultira softverom niskog rizika.

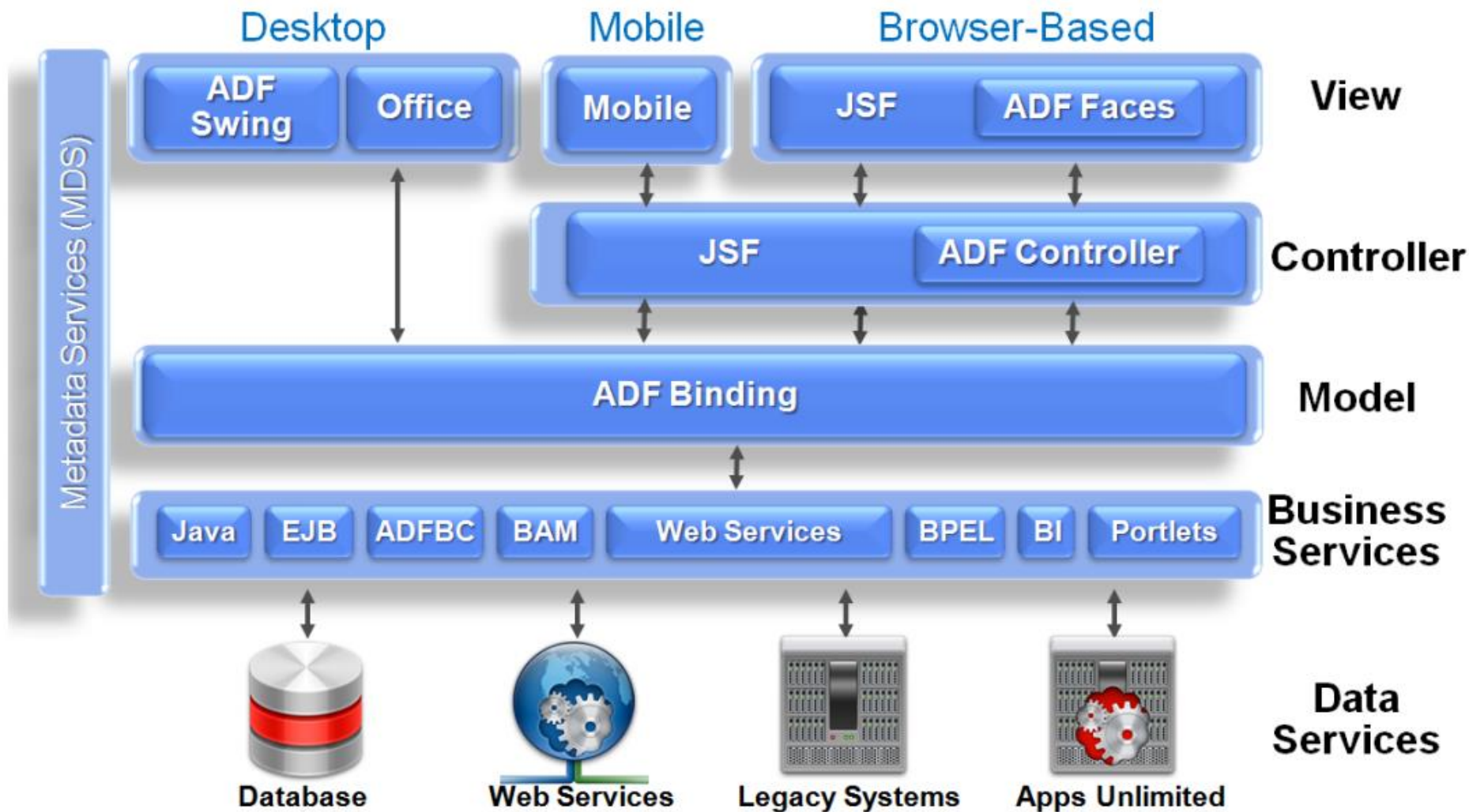
## Proces testiranja

- Bitne faze prilikom procesa testiranja su sledeće:
  - **PLANIRANJE TESTA**
    - Izrada plana testiranja, sa aspekta procesa testiranja, ima isti značaj kao i specifikacija dizajna sa aspekta dizajna.
  - **PRIPREMA TEST PODATAKA**
    - Osnovni problem je, uz poznat domen podataka i definisano očekivano ponašanja, kako generisati skup test podataka koji poseduje veliku verovatnoću otkrivanja još neotkrivenih grešaka?
  - **IDENTIFIKACIJA GREŠAKA (ERRORS)**
    - Provera svih test-izlaza i dokumentovanje rezultata. Ako se ne izoluje nijedna greška PROVERITI KOMPLETNOST SKUPA TEST-PODATAKA!
  - **TESTIRANJE PO KOREKCIJI** (regression testing)

# Tehnike testiranja u različitim fazama projekta

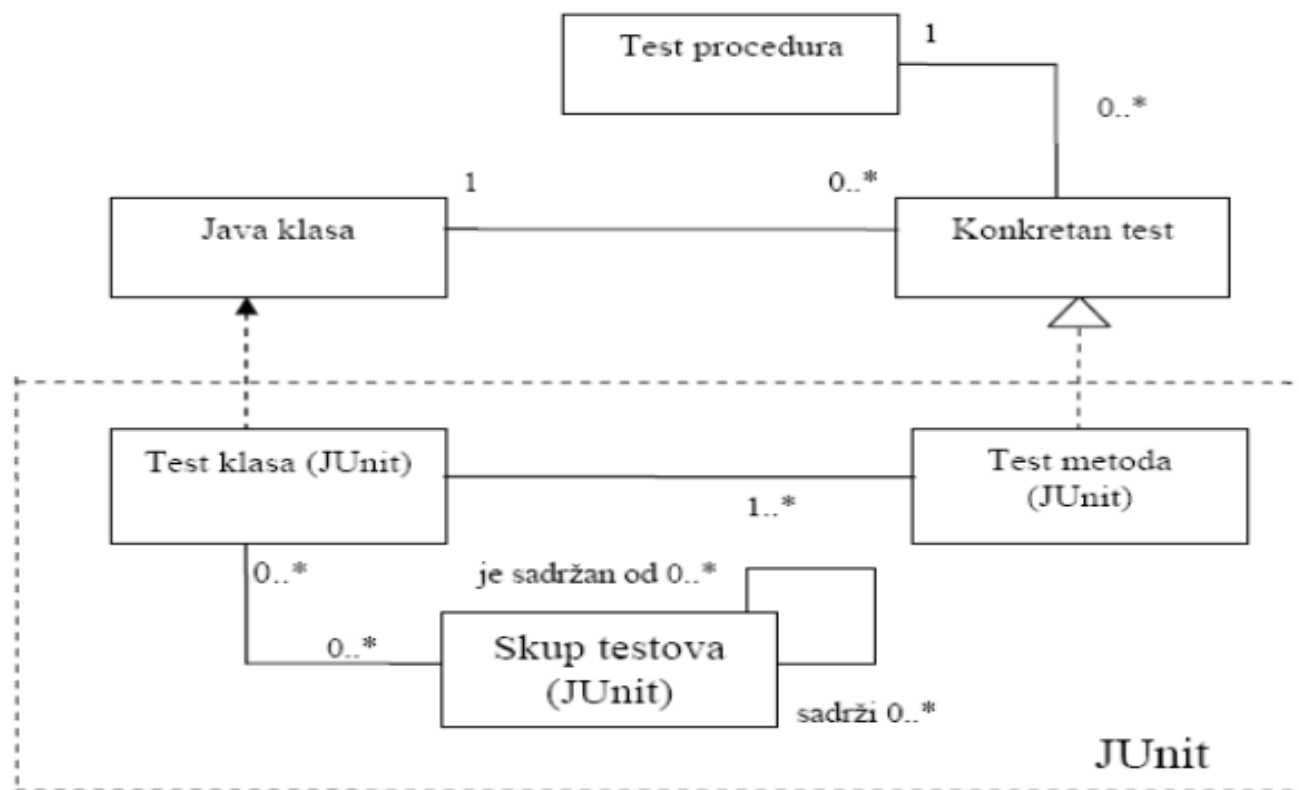


# Oracle ADF



## Testiranje ADF poslovnih komponenti - JUnit

- JUnit je open source okvir za testiranje
- Napisan za Java programski jezik
- Sastoji se iz sledećih gradivnih elemenata:





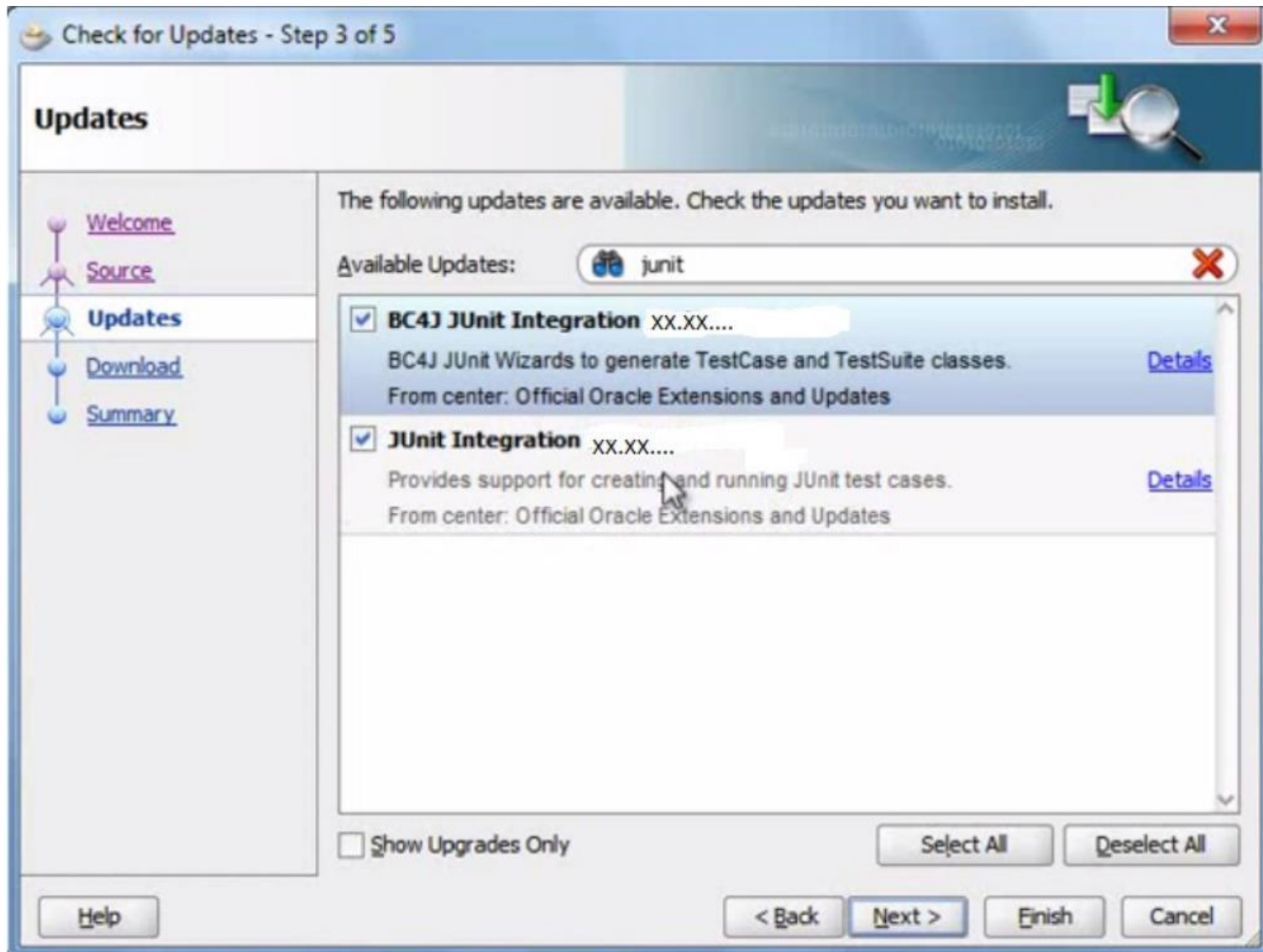
## Testiranje ADF poslovnih komponenti

- JUnit testiranje je veoma teško primenljivo na ADF aplikacije zbog:
  - spoljnih zavisnosti (baza podataka)
  - izolovan kod ne može lako biti izvršiv (naročito kod entity object-a, view object-a, ADF Facesa...)
- Ovo ne znači da treba odustati od Unit testiranja
- Jdeveloper proširenje za testiranje BC je korisno za generisanje skeletona testnih klasa.
- Testovi se startuju iz AM, kada se koristi proširenje
- Generisani test ne testira datu funkcionalnost nego zapravo generiše text fixture ( konfiguracija i inicijalizacija neophodnih objekata)
- Veoma je teško ili nije moguće direktno testirati entity object-e.

## Testiranje ADF poslovnih komponenti

- Testirati kod koji je ručno napisan.
- Ne testirati kod koji je Oracle napisao, izuzev kada su redefinisane postojeće funkcionalnosti Oracle ADF-a
- Praktični saveti za testiranje ADF poslovnih komponenti:
  - pravila validacije
  - podrazumevane vrednosti
  - klase koje su proširenja (testiraju se proverom pozadine klasa koje nasleđuju)
  - Servisne metode aplikativnog modula
  - Bilo koji kod koji se piše u EO/No Java klasama

## JUnit instalacija u JDeveloper-u



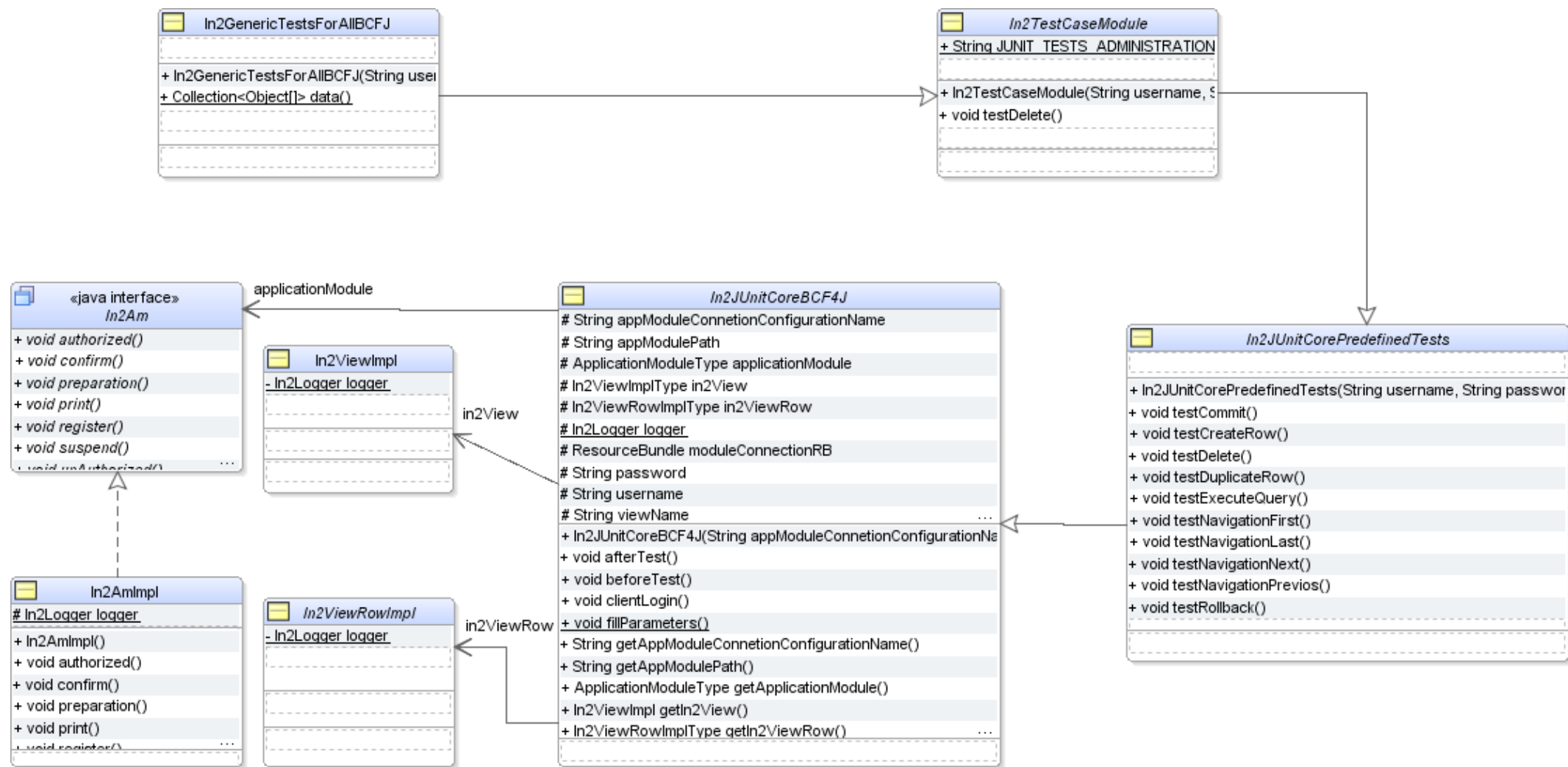
## Generičko testiranje

- Izgradnja vlastitog frameworka za testiranje ADF BC
- Pojednostavljena struktura
- Izbegavanje pisanja istog koda
- Olakšano održavanje
- Povećanja produktivnosti testiranja
- Fokus developera/testera na sam proces

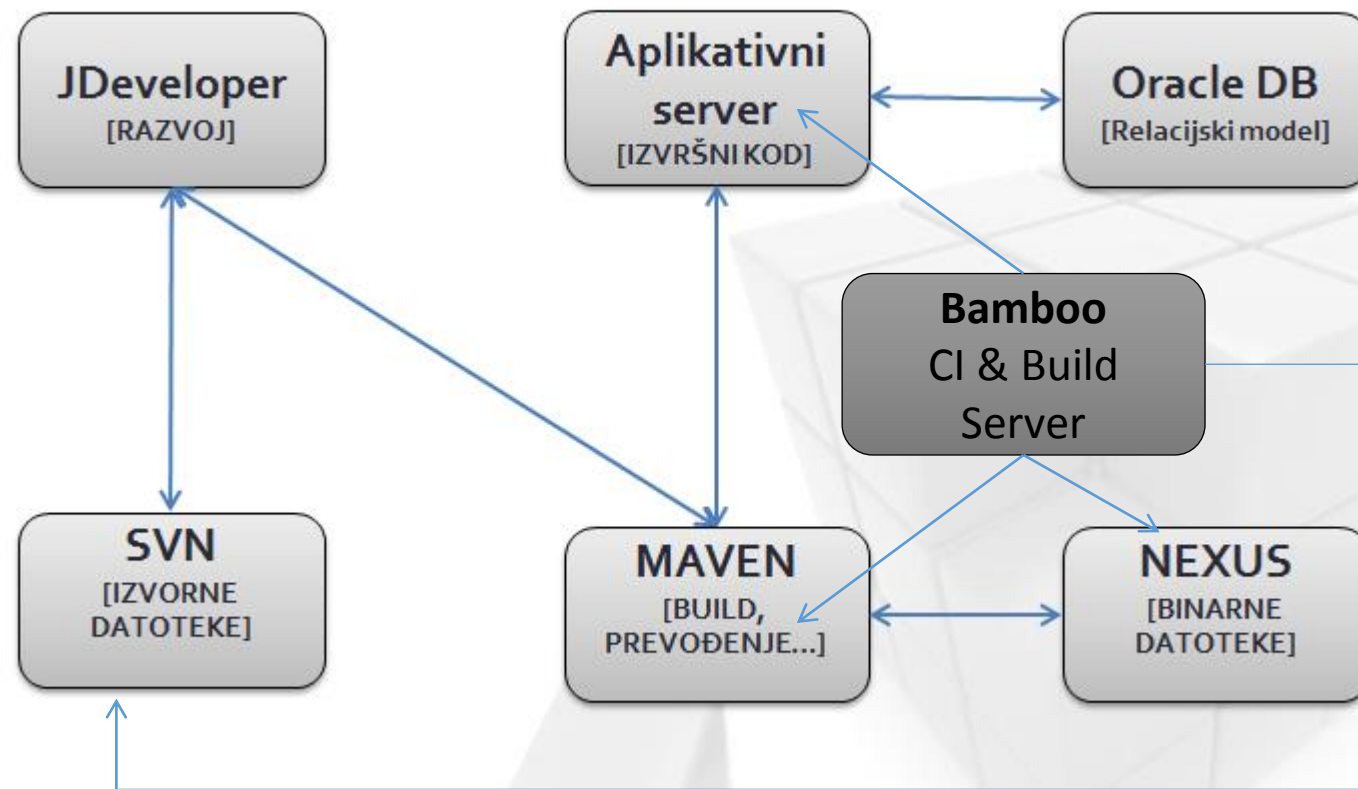
# Generičko testiranje



# Generičko testiranje – Konceptualni model



## Testiranje u funkciji kontinualne integracije



# Demonstracija na realnom case-u



# Pitanja



Hvala

[www.in2.rs](http://www.in2.rs)

[jovan.stevanovic@in2.rs](mailto:jovan.stevanovic@in2.rs)