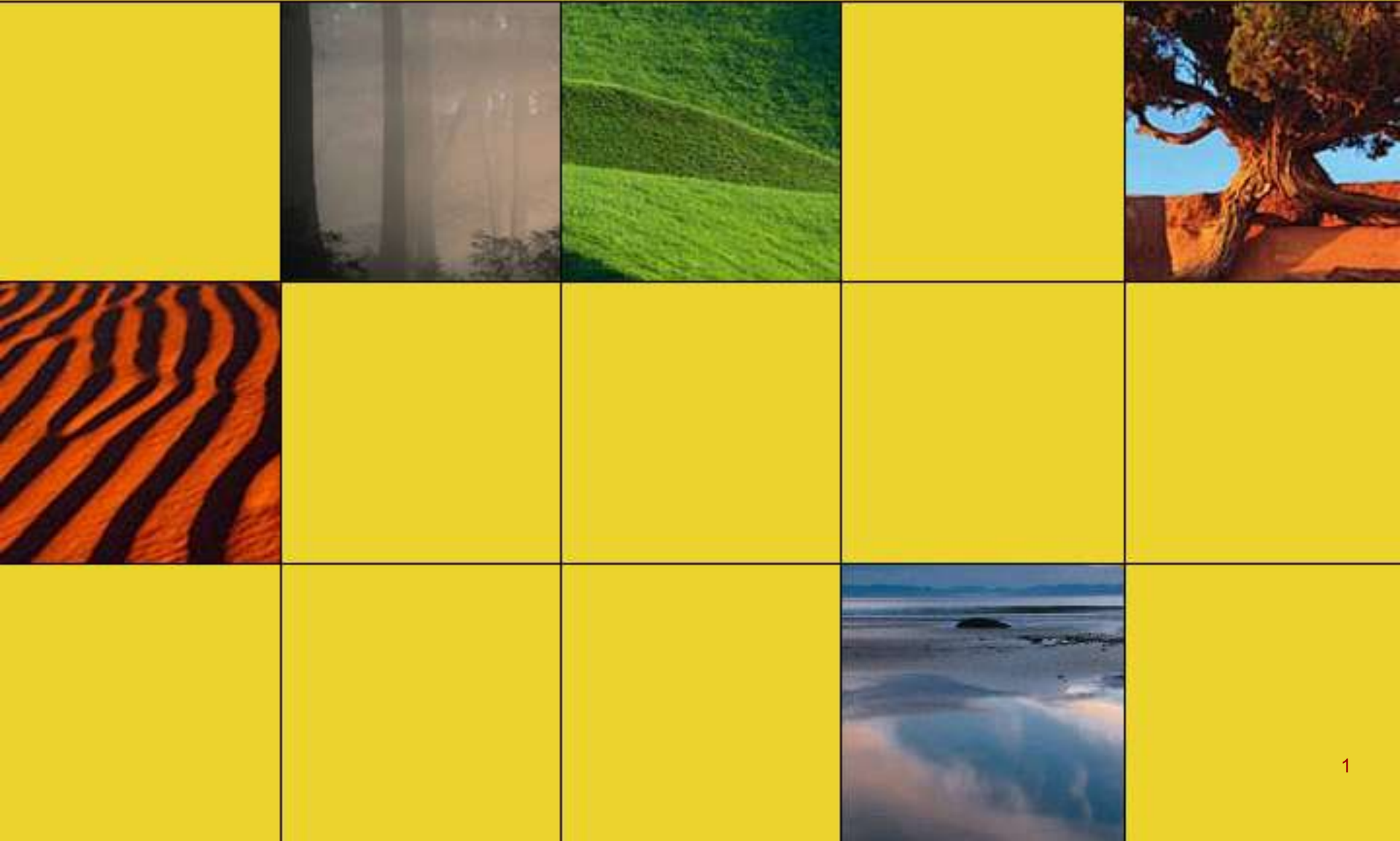


Open Source projekt: CVS query language

Siniša Tomić

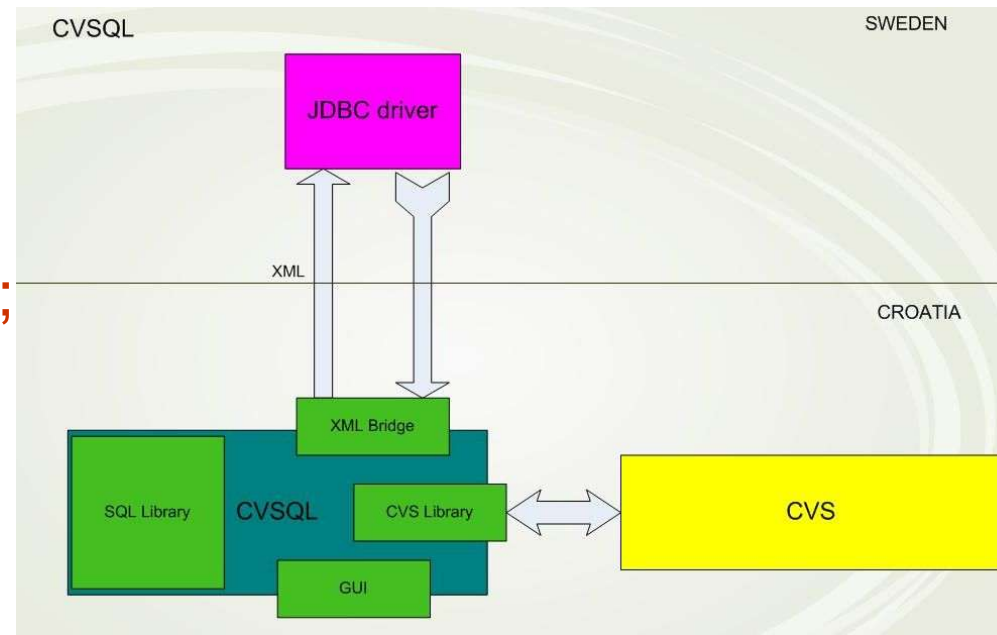


Što je CVS SQL?

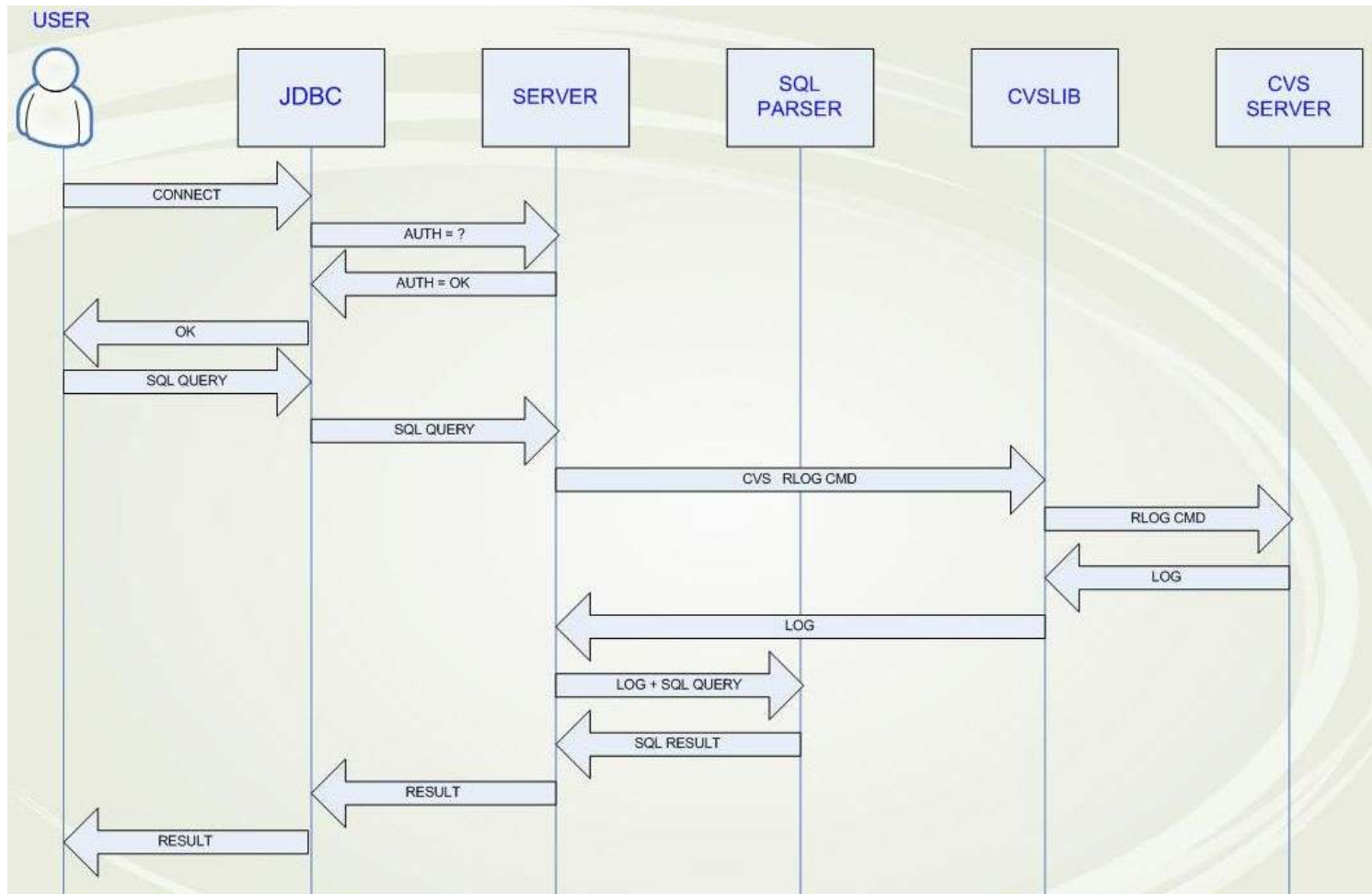
- CVS – Concurrent Versions System
- SQL – Structured Query Language
- CVS + SQL => CVS SQL 😊



- Želite li moći ovo?
 - SELECT dir, count(*)
FROM changes
WHERE author ='avril';
 - ; 154;
 - configs; 2;
 - img; 2;

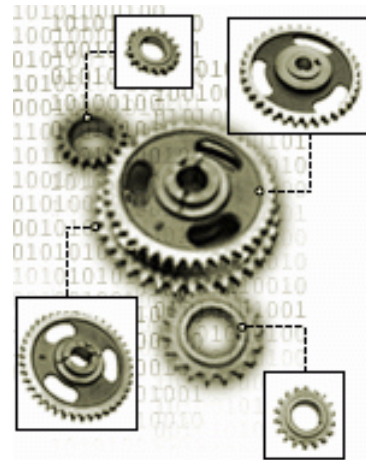


Kako to uopće radi?



Kako brzo napraviti program

- Open source nudi par zanimljivih rješenja
 - Pronaći gotovo rješenje i promijeniti ime autora
 - autorska prava
 - održavanje
 - gnjev originalnog autora
 - Pronaći dva gotova programa i spojiti u jedan
 - održavanje
 - spajanje dva nekompatibilna programa
 - Iskoristiti postojeće dijelove – ne otkrivati toplu vodu
 - Vrijeme potrebno za shvaćanje gotovog rješenja
 - Dokumentacija, testiranje...
 - Još uvijek najbolje, ali samo za male dijelove programa



Kako napraviti program

- Dobro specificirati što se želi napraviti
- Izabrati jezik pisanja programa
- Testirati program
- Napisati dobru dokumentacija
 - korisničku
 - tehničku
- Projektni pristup



Specifikacija

- Ako se zna što autor želi lakše će pronaći suradnike
- Izbjegavanje podjele poslova “u letu”
- Unaprijed se zna kada treba odustati – najteže
 - Projektni pristup
 - Niti svi Open source projekti ne moraju uspjeti
 - većina ih niti ne uspije, ali barem možemo uštedjeti vrijeme ako to unaprijed znamo 😊
 - Možda netko drugi neće napraviti istu grešku?
- Bolje planiranje budućnosti programa

Jezik programiranja

Metode odabira jezika:

- za ovaj sam čuo jučer – izgleda mi kul!
 - Ako je i stvarno dobar šanse da će prvi program u njemu biti dobar su vrlo male 😊
- za ovaj imam 14 knjiga – da ih iskoristim
 - Ako su knjige do sada bile nepročitane, vjerojatno će takve i ostati 😊
- u ovom sam programirao u srednjoj školi
 - One vježbe bi možda i mogli ponoviti, ali nešto više od toga?
- ovaj koristim svaki dan, i možda to može
 - Bolje je prvo provjeriti da li stvarno to i može
- u ovom već ima nekoliko sličnih programa i znam ga
 - Vrlo dobar izbor



Kako se provjerava program?

Najčešći načini testiranja

- ona tri prva primjera? Sada rade => sve radi
- onaj tip od jučer mi je rekao da je nešto probao i da mu sada radi, iako ga do jučer nije niti instalirao
- prošli puta mi je radio, mora i sada
- Meni radi™

Posljedica takve provjere?

U velikom broju slučajeva program zapravo ne radi
Već ste čuli?

- sve radi osim onih par najčešćih slučajeva
- sve mi radi dobro, ali mi OS povremeno ubije program
- nije moj program kriv, korisnik ga krivo koristi
- kako to misli “on nema backup”?
- zašto bi ga program išta pitao kada se klikne na “obriši sve”?

Problemi s jezicima

Svaki jezik pruža svoje veselje:

- C/C++
 - jeste li oslobodili memoriju
- Java
 - XML parser ima problema s neispravno napisanim XMLom – korisnika to ne zanima
- PHP
 - zašto mi je varijabla prazna? (hint: `$file != $flie`)
- Basic
 - goto is my friend 😊

Problemi s OS-om

Ljepota portabilnosti:

- Linux
 - ma rade meni threadovi, samo ...
- Windowsi
 - gdje je ono nestao fork()?
- Symbian
 - sad ću ja doći do podataka, ček da shvatim u kojoj su klasi
- PalmOS
 - ok, gdje su datoteke, ne treba mi baza



Kvalitetno testiranje

- Definitivno imati više od 3 testna primjera
- Pokušati odabrati velik broj ispravnih primjera
- Na ispravnim primjerima napraviti male izmjene i pokušati ponovo – neispravni primjeri daju više informacije
 - primjer: tekstualni unos SQL-a
 - SELET * FROM changes; -- ruši program
 - SELECT * FROM chnages; -- ne oslobađa mem.
 - SELECT * FROM changes; -- sve radi dobro -> program ne(!)
 - SELECT * FROM changes2; -- radi iako ne bi trebao – feature



Programsko testiranje

- Ručno isprobavanje kombinacija?
 - odustanite odmah, ionako ćete prije ili kasnije 😊
- Vrste programa
 - besplatni i komercijalni programi
 - provjera pisanja po memoriji
 - provjera ulaznih podataka
 - snimanje i ponavljanje rada s vašim programom
 - najčešće najbolje napraviti i vlastiti
 - provjera rezultata na svim platformama
 - “inside” informacije i znanje



Dokumentacija

Dobar kod ne treba dokumentirati

- Legenda o kodu koji nije trebalo dokumentirati
- Zar stvarno planirate imati toliko kratak kod
 - rijetko kada će i ostati toliko kratak
 - vrijeme potrebno za proučiti 250000 linija je veće od vremena potrebnog za pogledati 5 dijagrama
- Nakon nekoliko ispravaka bugova neće biti više dobar
 - ako i preživi ispravke bugova, dodavanje novih mogućnosti sigurno neće 😊
- Svakome je jasno zašto se nakon getNextParsedChunk mijenja vrijednost varijable leftItem
 - pa i nakon što ćete promijeniti programski jezik i stil programiranja

Pitanja?

- Tko očekuje da moj program zadovoljava te kriterije?
 - vjerojatno korisnik, ali to ćete ga morati pitati sami 😊