

# Mittefunktsionaalsete nõuete staatiline analüüs

Ervin Hülpp

Tallinn 2009

# Sisukord

- **Mis** see on
- **Miks** seda teha
- **Kuidas** seda teha
- **Mida** oleme teinud
- **Mida veel** teha

# Mis see on?

- **Mittefunktsionaalne vs funktsionaalne**
  - KUIDAS vs MIDA; nt. jõudlus, kasutatavus, turvalisus, ühilduvus
  - Nõuded üldisemad, rakendusest ja valdkonnast vähem sõltuvad
- **Staatiline vs dünaamiline**
  - Compile vs runtime, off-line vs on-line processing
  - Rohkem aega (laiem, sügavam vaade), vähem infot (valehäired)
  - Uued keeled dünaamilisemad ent staatilised võimalused olemas
- **Analüüs vs testimine**
  - Abstract vs concrete interpretation, type vs value checking
  - White-box vs black-box testing (vananenud?)

“My God, it’s full of stars!”  
Bowman (“2001”)

# Miks seda teha?

- **Tasub ära**

- Vastused silme ees, kui oskad vaadata

“We didn’t have to replicate the problem. We understood it.”

Linus Torvalds

- Võib vaadelda korraga tervet vigade klassi, mitte üksikviga
- Võib tegeleda korraga terve süsteemiga (mudel, koodi- või andmebaas), mitte üksiku stsenaariumi, faili või klassiga

- **Huvitav**

- Võib näha keerulist süsteemi Suure Pildina

“I don't even see it encoded anymore,  
all I see is blonde, brunette, redhead.”

Cypher (“The Matrix”)

# Kuidas seda teha?

- ... ehk "Mittefunktsionaalsete nõuete staatilise analüüsi" mittefunktsionaalsete nõuete staatiline analüüs
- Võimalikult täpselt, kiiresti ja odavalt
- Arvesta juba ette võimalikke riske
  - Nõuded erineva abstraktsuse tasemega (multi view / lang)
  - Nõuded muutuvad ajas (paindlikkus, lihtsus)
  - Analüüs ei õnnestu (incomplete src / parser)
  - Valehäirete ja erandite haldus (id, mark, filter)

# Mida oleme teinud?

- Olemasolevate vahendite uurimine (lint, PMD, FindBugs, FxCop, CodeSonar, Prevent)
- Meetrikad (wc -l, McCabe)
- Stringiotsing (grep, find, Perl, Google Code)
  - Case, whitespace, REM, multiline, abbrev, permuted
- Leksika- ja süntaksianalüüs (lex, yacc, ANTLR)
  - Sisuliselt kompilaatori / interpretaatori front-endi koopia
- Data Dictionary, XREF analüüs
  - IO, access & integrity rules, dependencies
- Mustrid (XQuery <http://www.w3.org/TR/xquery/>)
  - Positiivsed (coding std, normal forms, patterns, white list)
  - Negatiivsed (vulnerabilities, code smells, antipatterns, black list)

# Näide

- Näita 'Date' lõpuga parameetreid (EndDate, TransactionDate)

*//parameter[ends-with(./@value,'Date')]*

- Näita andmebaasi poole pöördumisi, kus kirje lukustus on määramata ning tegemist pole ajutise tabeliga

```
FOR $S in //(find_statement|for_statement)[count(./record/lock) = 0]  
LET $R := $S/record/@value  
WHERE every $TMP in //temptable_statement/record  
      satisfies lower-case($TMP/@value) != lower-case($R)  
ORDER BY $R  
RETURN $S/@line
```

# Mida veel teha?

- Täiendada analüüsi (suurem täpsus)
  - CFG (dead code, temporal logic), DFG (data leaks & misuse), DSM (architecture / dependency mgmnt, layering & cycles)
  - Muudatuse terviklikkus (musteri genereerimine koodist)
  - Dünaamiline info (profiler, log, trace)
- Kasutada toetustehnoloogiaid (suurem kiirus, odavus)
  - Editor plug-in (JIT)
  - CI server (check-in, daily / nightly build)
  - Andmebaas (ad-hoc query)
  - AOP

# Täna!

- Küsimused
- Kommentaarid
- Arutelu