



Staattinen testaus

Luento 5
Antti-Pekka Tuovinen

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Faculty of Science
Department of Computer Science

www.cs.helsinki.fi

26 March 2013

1



Oppimistavoitteet

- Katselmointien käyttö vikojen löytämiseen
- Ohjelmakoodin ja spesifikaatioiden staattinen analyysi

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Faculty of Science
Department of Computer Science

www.cs.helsinki.fi

26 March 2013

2



Staattinen testaus

- Testauksen kohdetta ei suoriteta, vaan sitä tutkitaan ihmismielen ja analyysityökalujen avulla
 - Mitä tahansa kehitysprojektin tuotosta voidaan tutkia manuaalisesti
- Tavoitteena on nostaa tutkittujen artefaktien (dokumentit, koodi) laatua ja poistaa vikoja mahdollisimman aikaisessa vaiheessa
 - Ongelmien ennaltaehkäisy



Katselmoinnit



Katselmoinnit

- Käytetään ihmisten analyyttisiä kykyjä mutkikkaiden asioiden arviointiin ja tarkastamiseen
 - *Asiantuntijat* lukevat huolellisesti katselmoitavan dokumentin ja yrittävät ymmärtää sen
 - Löydösten läpikäynti ja johtopäätösten tekeminen on *ryhmätyötä*, jossa monet eri näkökulmat yhdistyvät



Katselmoinnit

- *Katselmointi* (review) on yleistermi ihmisten tekemälle kaikenlaisten *dokumenttien* arvioinnille
 - Sopimukset, vaatimusmäärittelyt, ohjelmakoodi, testaus suunnitelmat, manuaalit jne...
- *Tarkastus* (inspection) on erityinen muodollinen katselmointi, jolla on omat sääntönsä
- Katselmoijat ovat yleensä saman työyhteisön jäseniä, joten katselmoinneista käytetään usein engl. termiä *peer review*



Katselmoinnit

- Katselmointi voidaan tehdä *heti*, kun dokumentti/koodi on valmis
 - Viat ja epäjohtonmukaisuudet löydetään aikaisin, jolloin niiden vaikutukset eivät kertaudu projektin myöhemmissä kehitysvaiheissa
 - ...edellyttäen, että katselmoitava dokumentti valmistuu käyttötarkoitukseensa nähden ajoissa
 - Saadaan eri näkökulmia mukaan aikaisessa vaiheessa (esim. suorituskyky, tietoturva)
- "Phase exit review"



Katselmoinnin hyödyt

- Vikojen vähentäminen, kuten testauksessa yleensä
- Aikainen vikojen löytäminen vähentää tarvetta analysointiin ja korjauksiin myöhemmin projektissa
- Kehitysaika lyhenee
- Koska ohjelmistossa on vähemmän vikoja, dynaamista testausta tarvitsee tehdä vähemmän (uudelleentestaus, regressiotestaus)
- Kustannukset laskevat koko elinkaaren ajalla



Katselmoinnin hyödyt

- Vähemmän käytön aikana havaittavia vikoja (failure rate)
- Organisaatio oppii
 - Tekijät oppivat parantamaan työskentelyään
 - Tieto tulevasta katselmoinnista saa tekijän tuottamaan selkeän dokumentin ja huomaamaan itsekin puutteita
 - Katselmointiin osallistuva ryhmä muodostaa yhteisen näkemyksen dokumentin kuvaamasta asiasta ja tuntee yhteistä vastuuta siitä
- Kustannukset 10-15% projektin kuluista, säästö 14-25% (katselmoinnit mukaan laskien)



Katselmoinnin kulku

- Prosessikaavio ja yleiskuvaus IEEE 1028-1997 Standard for Software Reviews –standardin mukaan (kurssikirjan teksti perustuu samaan)
 - http://www.the-software-experts.de/e_dta-sw-test-inspection.htm



Katselmointikokouksen säännöt

1. Kokous saa kestää korkeintaan kaksi tuntia
2. Valvoja (moderator) voi keskeyttää kokouksen, jos joku katselmoija (asiantuntija) ei tule paikalle tai on puutteellisesti valmistautunut
3. Arvioinnin kohteena on katselmoitava dokumentti, ei sen tekijä
4. Valvoja ei saa olla myös katselmoija
5. Annettujen ohjesääntöjen (guideline) ulkopuolelle jäävistä tyyliseikoista ei keskustella
6. Kokouksen tehtävä ei ole kehittää ratkaisuja löydettyihin ongelmiin



Katselmointikokouksen säännöt

7. Jokaiselle katselmoijalle on annettava riittävästi aikaa löydöstensä esittämiseen
8. Kaikkien pitää olla yhtä mieltä pöytäkirjaan kirjattavista asioista
9. Löydöksiä ei saa kirjata käskyinä tekijälle tehdä määrättyjä muutoksia
10. Löydökset on luokiteltava niiden vakavuuden mukaan
11. Katselmointiryhmä esittää joko dokumentin hyväksymistä tai hylkäämistä
12. Kaikki allekirjoittavat lopuksi pöytäkirjan



Katselmointien ongelmia

- Asiantuntijoiden saatavuus
- Katselmointeihin ei varata tarpeeksi aikaa tai tulee muuten kiire, jolloin katselmoinnin teho kärsii
- Väärien katselmoijien valinta
 - Ei ehdi, ei osaa, ei kiinnosta
- Dokumentaation puutteet
- Johdon puuttuva tuki
 - Resurssointi, tulosten hyödyntäminen



Katselmointityyppejä

- Edellä kuvattu formaali katselmointi on suhteellisen raskas prosessi (aika & resurssit)
- Tekniset katselmoinnit jaetaan raskautensa mukaan seuraaviin ryhmiin
 - Läpikäynnit (walkthrough)
 - Tarkastukset (inspection)
 - Tekninen katselmointi (technical review)
 - Epämuodollinen katselmointi (informal review)



Läpikäynti

- Epämuodollinen katselmointi, jossa tekijä esittelee katselmoinnin kohteen ryhmän arvioitavaksi tavoitteena
 - Poikkeavuuksien löytäminen (anomaliat)
 - Laadun parantaminen
 - Vaihtoehtoisten ratkaisujen löytäminen
 - Standardien ja spesifikaatioiden mukaisuuden arviointi
- Tärkeimpien käyttötapausten simulointi
- Tekijä toimii moderaattorina
- Sopii pienille tiimeille, ei vaadi valmistautumista eikä jälkitöitä



Tarkastus

- Kaikkein muodollisin katselmointi
- Yksityiskohtaiset säännöt ja tarkistuslistat
- Huolellisesti valitut osallistujat, jotka voivat tarkastella asiaa eri näkökulmista
- Keskittyy etukäteen päätettyihin asioihin tarkastuksen kohteessa
- Perusteellinen valmistautuminen ennen katselmointikokousta



Tekninen katselmointi

- Pääasioina tarkastuksen kohteen
 - Verifiointi sen spesifikaation suhteen
 - Soveltuvuuden arviointi käyttötarkoitukseensa
 - Standardien mukaisuuden arviointi
- Katselmoijien valintaperusteena tekninen pätevyys
- Myös projektin ulkopuolelta on hyvä olla joku asiantuntija mukana
- Suurin työ tehdään valmistautumisvaiheessa, katselmointikokouksessa kirjataan löydökset
- Tekijä itse ei osallistu kokoukseen



Epämuodollinen katselmointi

- Katselmoinnin "kevytversio"
- Usein katselmoinnin kohteen tekijä itse kutsuu osallistujat ja pyytää heiltä kommentit tiettyyn ajankohtaan mennessä
- Katselmointikokousta ei välttämättä järjestetä, vaan kukin osallistuja toimittaa havaintonsa tiedoksi tekijälle
- Yleinen katselmoinnin muoto, jossa on matala osallistumiskynnys
 - Pariohjelmointi, "kaveritestaus"



Huomioita

- Katselmointia on monenlaista ja eri tyyppien väliset rajat ovat häilyvät
 - Samoja termejä käytetään eri tarkoituksissa
- Katselmointikäytännöt räätälöidään organisaation tarpeiden ja katselmoinnin kohteen mukaan
 - Virtuaalikatselmoinnit ja on-line kokoukset helpottavat hajautettujen organisaatioiden toimintaa
- Kun kehittäjät tutkivat myös toistensa työn tuloksia, vikoja yleensä huomataan enemmän
 - Pariohjelmointi on tavallaan jatkuvaa parikatselmointia



Staattinen analyysi



Staattinen analyysi

- Kuten katselmoinneissa, tarkoituksena on löytää vikoja tai vioille alttiita kohtia tarkasteltavasta dokumentista
- Analysointiin käytetään *automaattisia työkaluja* eli algoritmisia menetelmiä käsin tehtävän analyysin sijaan
- Analyysityökaluja voidaan käyttää myös mittausten tekemiseen kohteesta laadun arvioimiseksi



Staattinen analyysi

- Dokumentin analysoitavuus edellyttää, että se on koodattu jollain formaalilla kielellä
 - Syntaksi ja semantiikka ovat analysoitavissa algoritmisesti
- Esimerkiksi UML-mallit ja erilaiset XML-pohjaiset kuvaukset voivat sopia analysoitaviksi
- ...ja tietysti ohjelmakoodi



Staattinen analyysi

- Ohjelmistokehittäjillä on käytettävissään erilaisia analysaattoreita, joilla voi kätevästi tarkistaa, noudattaako koodi yleisiä tyylisääntöjä ja koodaussuosituksia
- Staattinen analyysi voi paljastaa epäilyttäviä kohtia ja vaarallisia ohjelmarakenteita, joita dynaaminen testaus ei paljasta
- Toisaalta staattinen analyysi ei voi koskaan löytää kaikkia vikoja
 - Se vaatisi ohjelman kaikkien mahdollisten tilojen simulointia



Staattinen analyysi

- Staattinen analyysi sopii hyvin yhteen katselmointien kanssa
 - Kaikki riittävän formaalit dokumentit/koodi kannattaa ajaa analysaattorin läpi katselmointiin valmistauduttaessa



Staattinen analyysi

- Staattisella analyysillä voidaan löytää seuraavanlaisia potentiaalisesti ongelmallisia rakenteita
 - Syntaksivirheet
 - Poikkeamat koodauskäytännöistä ja –standardeista
 - Kontrollivuon säännöttömyydet
 - Datavuon säännöttömyydet
 - Eräät tietoturvan kannalta vaaralliset koodirakenteet



Kääntäjä analyysityökaluna

- Kääntäjä löytää ohjelmasta syntaksivirheet ja muut ohjelmointikielen kieliopin mukaan virheelliset ilmaisut
- Näiden lisäksi kääntäjä tekee muitakin tarkistuksia ja tuottaa paljon informaatiota
 - Tyypitarkistukset (staattinen tyypitys, tyyppien päättely)
 - Määrittelemättömät ja alustamattomat muuttujat
 - Saavuttamaton koodi
 - jne...



Datavuoanalyysi

- Analysoidaan datan käyttöä eri suorituspoluilla ohjelmayksikön läpi
- Voidaan havaita säännöttömyyksiä (anomaly), jotka voivat johtaa virheelliseen suoritukseen – tai sitten ei
- Tarkastellaan esimerkiksi muuttujien käyttöä ohjelmassa
 - *Defined (d)*
 - *Referenced (r)*
 - *Undefined (u)*



Datavuoanalyysi

- Anomaliat
 - **UR** – arvoltaan määrittelemättömän (u) muuttujan arvoa käytetään (r) jollain suorituspolulla
 - **DU** – muuttujalle annetaan arvo (d), jota ei käytetä kertaakaan ennen sen muuttumista määrittelemättömäksi (u)
 - **DD** – muuttujalle annetaan uusi arvo ilman, että sen edellistä arvoa on välillä käytetty



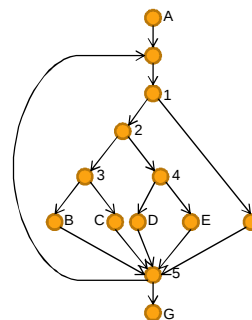
Datavuoanalyysi

- Kurssikirjan esimerkki sivu 99
- Datavuoanalyysin löytämät anomaliat eivät välttämättä johda ohjelman virheelliseen käyttäytymiseen, mutta löydökset kannattaa joka tapauksessa tutkia



Kontrollivuoanalyysi

- Ohjelman kontrollivuoverkosta voi helposti nähdä riskialttiita rakenteita
 - Hypyt pois silmukan keskeltä
 - Monimutkaiset, sisäkkäiset valintarakenteet
 - Monta erilaista tapaa poistua rakenteesta (return)





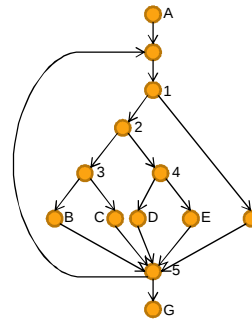
Kompleksisuusmetriikat

- MaCabe – cyclomatic number

$$v(G) = e - n + 2$$

e = verkon kaarien lukumäärä

n = verkon solmujen lukumäärä



Kompleksisuusmetriikat

- McCabe:n luku voi osoittaa kohtia ohjelmasta (esim yksittäisiä metodeja > 10), joiden testaukseen kannattaa kiinnittää enemmän huomiota
- Metriikkaa on myös arvosteltu siitä, että se ei ota huomioon valintausekkeiden ehtojen monimutkaisuutta
- Muunkinlaisia metriikoita on kehitetty ja niiden tilastollista yhteyttä löydettyihin vikoihin on analysoitu
 - Niinkin yksinkertainen mittari kuin koodirivien määrä (esim metodin pituus) korreloi hyvin vika-alttiuden kanssa