Laadunvarmistussuunnitelma

Ohjelmisto on kevyt käyttäjäkeskeinen verkkosovellus. Olemme pyrkineet valitsemaan laatutekijät sellaisiksi, että niiden kautta voidaan keskittyä pitämään ohjelmiston yksinkertaisena ja virheettömänä, sekä erityisesti huomioimaan sen käytettävyyttä. Valitsemamme laatutekijät ovat: verifioitavuus, ylläpidettävyys, käytettävyys ja joustavuus.

Laatutekijät, niiden osatekijät ja mittarit:

Verifioitavuus: Miten helposti ohjelmiston toiminta on varmennettavissa.

- Pituus, Ohjelmiston pituus on tarkoitus pitää mahdollisimman lyhyenä, mitä vähemmän koodia, sen helpompi sen helpompi sen toiminta on varmentaa
    - Rake stats,
- Virheiden löydettävyys, käytetään mittareita jonka tekevät virheiden löytämisen mahdollisimman tehokkaaksi ja täydelliseksi.
    - Flog
    - RCov
    - Cucumber
    - RSpec

Ylläpidettävyys: miten työlästä on etsiä ja korjata virhe ohjelmistosta

- Kompleksisuus, Jos ohjelmiston rakenne pysyy yksinkertaisena, on virheiden etsiminen ja korjaaminen helpompaa
    - Saikuro
    - Flog
    - Flay
    - Roodi
    - Reek
- Pituus, mitä vähemmän koodia, sen helpompaa sieltä on löytää virheitä
    - Rake stats
- Virheiden löydettävyys, Miten virheiden löytämistä on helpotettu
    - Flog
    - RCov
    - Cucumber
    - RSpec
    - Bugilogi

Käytettävyys: miten käyttäjäystävällinen ohjelmisto on

- Käytön tehokkuus, Kuinka paljon vaiheita toimintojen käyttäminen vaatii. Vähemmän vaiheita, parempi käytettävyys.
    - Cucumber-featuret
    - Käyttäjätestit

Joustavuus: miten työlästä on muuttaa ohjelmiston toiminnallisuutta

- Kompleksisuus, Ohjelmiston monimutkaisuus vaikeuttaa sen toiminallisuuden muuttamista.
    - Saikuro
    - Flog
    - Flay
    - Roodi
    - Reek

Käytettävät tekniikat ja mittarit:

- saikuro: metodien syklomaattinen kompleksisuus, mittaa lineaarisesti itsenäisten polkujen summaa.

Kertoo joka metodille syklomaattisen kompleksisuuden, arvon tulisi olla < 11.

- flog: Miten vaikeata on testata.

antaa ns. ABC-arvon koodista, Assigments, Branches, Calls. Sjoitusoperaatioista, haarautumisista ja kutsuista. Arvon tulisi olla < 40

Tällä voidaan myös nähdä jos jokin metodi on selkeästi monimutkaisempi kuin jokin muu.

- flay: duplikaatiot, refaktoroitavia kohteita

Tutkii koodista samankaltaisuuksia arvoissa, nimistausssä ja rakenteissa. Kertoo onko syytä refaktoroida samankaltaista koodia.

- rcov: code coverage, missä määrin koodi on testattu.

Kuinka paljon koodista testit kattavat. Tavoitteeksi on päästä 99% prosenttiin.

- roodi ja reek: design ongelmat, valittaa jos koodi on huonosti laaditu.

Ennenkaikkea katselmoinnin apuväline, reek mittaa koodin "haisevuutta" (huonoa ja epäselvää koodia)

Roodi mittaa koodin huonosta suunnittelusta aiheutuvaa kompleksisuutta sekä joitakin tyypillisiä koodivirheitä

- rake stats: loc, test loc, niiden ratio + muita statistiikkoja moduleittain

Koodirivien määrää, testirivien määrä, koodin ja testien suhde. Tällä voidaan myös huomata jos jokin metodi on merkittävästi muita laajempi.

- cucumber: BDD mukainen ohjelmointi.

Selväkielisiä testitapauksia, tehdään testit kaikille käyttötapauksille ja kaikki testitskenaariot saatava suoritettua onnistuneesti.

- hoptoad: etsii virheitä ohjelmistoVerifioitavuus toteutuu kun flogin tulos on: <= 40.

Kaikki ajonaikaiset virheet raportaidaan hoptoadiin, Esim asiakkaan testikäytössä virheet saadaan kirjattua automaattisesti.

- rspec: testaamisen ohjelmistokehys (BDD).

Mittaa testien läpimenoa.

- churn, Kuinka paljon koodia on muutettu.

Katselmoinnin apuväline

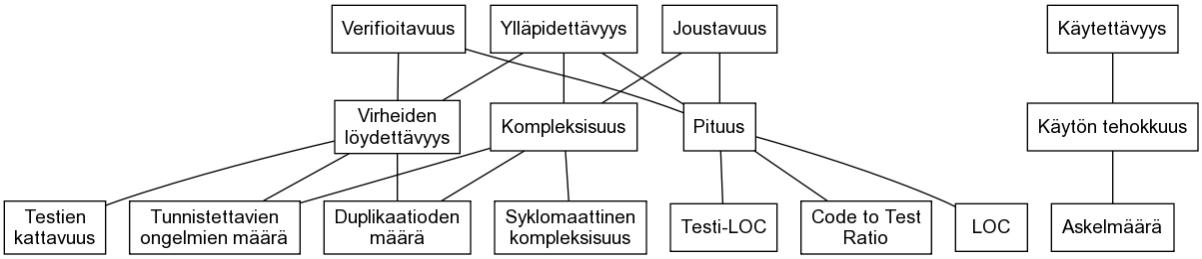
- Käyttäjätestit

Suoritetaan käytettävyystestausta koehenkilöillä.

- Bugilogi

Löydetyt bugit kirjataan logiin, miten ne voidaan toistaa ja muut tarvittavat tiedot.

Verifioitavuus toteutuu kun flogin tulos on: <= 40



Projektin laatu:

- projektin eteneminen ja laatu:
    - mitataan aikaansaadun koodin määrää suhteessa käytettyyn aikaan.
    - mitataan virheiden määrää suhteessa koodin määrään. (rake stats loc vs hoptoad).
    - kaikkeen työhön käytetty aika per sprint.
    - churnin mittaustulokset: miten koodia on muutettu, onko jouduttu refaktoroimaan samoja osia koodia moneen kertaan.

- Projekti on laadukas kun:
    - aikaansaatu koodi / käytetty aika on 50 riviä
    - virheiden määrä / loc <= 0.05
    - työhön käytetty aika / sprint >= 16
    - churnin tulokset arvioidaan katselmuksella

Riskienhallinta

Tiimi pyrkii tapaamisissaan jatkuvasti tunnistamaan ja analysoimaan projektia uhkaavia riskejä, suunnittelemaan vastatoimia sekä seuraamaan ja päivitettämään riskejä. Tunnistetut riskit kirjataan erilliseen riskidokumenttiin.

Projektissa käyettävät työvälineet:

- Parikoodaus
- Mittarit ja testit

Testausstrategia:

- BDD

Tarkistusstrategia:

- Viikoittainen koodin katsaus.

Dokumentit ja niiden saatavuus:

- Tarvittavat dokumentit ovat saatavilla projektille prustetusta Google Groupin tietokannasta, sekä
- Lähdekoodi on saatavilla Githubin repositoriosta.
    - Versionhallintaan käytetään Git:iä.

Ohjelmisto on laadukas

- Hoptoadin ilmoittamien virheiden määrä on: < k,
- testien kattavuus (rcov) on => 99%

- code-to-test ratio on välillä 1:1.1 - 1:1.3 ja
- kaikki testit menevät läpi (rspec & cucumber).
- sakuron tulos on: <= 11
- flay on <= 1000
- flogin ilmaisema testattavuus on <= 40
- katselmoinnissa hyväksytään reekin ja roodin ilmoittamat virheet (jos virheitä)s