# Verslag opdracht 1 Software Evolution

### Aanpak

Marco en Ivo hebben regelmatig via mail, WhatsApp en telefonisch overlegd. Tijdens deze contactmomenten is o.a. besproken op welke wijze de metrieken precies geïmplementeerd moeten worden. De verschillende activiteiten zijn op de volgende manier verdeeld:

|  |  |
| --- | --- |
| Activiteit | Wie |
| Complexity | Marco |
| Unit size | Marco |
| Duplication | Ivo |
| Volume | Ivo |
| Unit testing | Ivo |
| Conclusies trekken uit resultaten | Marco en Ivo |
| Verslag | Marco en Ivo |

### Duplication

In [[1]](#footnote-1) wordt er een oplossingsmethode voorgesteld om de metriek “Duplication” van een systeem te bereken waarbij blokken van 6 regels met elkaar worden vergeleken. Men moet zich realiseren dat indien men gebruik maakt van deze methode, er rekening mee moet worden gehouden dat alleen “concrete duplications” gevonden zullen worden. Stukken code die logisch gezien hetzelfde zijn, maar verschillende namen van bijvoorbeeld attributen gebruiken, zullen op deze manier niet als duplications worden aangemerkt. De implementatie verwijdert lege regels, blokcommentaar, regelcommentaar, import statements en voert ook een compressie uit door alle whitespace te verwijderen voordat met de duplicatie detectie gestart wordt. De resultaten voor de systemen HyperSQL en SmallSQL zijn de volgende:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| System | Metriek | Rank | System | Metriek | Rank |
| SmallSQL | 10.1% | - | HyperSQL | 14.7% | - |

Beide systemen hebben een rank van “-“ voor duplication. Zoals ook uit figuur 2 van Bijlage A blijkt, vlakt het histogram van HyperSQL niet snel af: Veel methoden hebben een duplication percentage in het bereik 10-25. Voor beide systemen geldt dat er meerdere bronbestanden zijn die zelfs een duplication hebben van 70 á 80%.

### Volume

De metriek “volume” wordt bepaald door het aantal regels te tellen in de broncode. Behalve de verwijdering van import statements, worden dezelfde preprocessing methode toegepast als bij de bepaling van de metriek “duplication”. De resultaten voor de systemen HyperSQL en SmallSQL zijn de volgende:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| System | Metriek | Rank | System | Metriek | Rank |
| SmallSQL | 24788 | ++ | HyperSQL | 163205 | + |

### Complexity

De metriek “complexity” wordt bepaald …..<Marco> . De resultaten voor de systemen HyperSQL en SmallSQL zijn de volgende:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| System | Metriek | Rank | System | Metriek | Rank |
| SmallSQL | 5 | -- | HyperSQL | 5 | -- |

<Marco> meer detail geven

### Unit size

De metriek “Unit size” wordt bepaald …..<Marco> . De resultaten voor de systemen HyperSQL en SmallSQL zijn de volgende:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| System | Metriek | Rank | System | Metriek | Rank |
| SmallSQL | 4 | - | HyperSQL | 5 | -- |

<Marco> meer detail geven

### Unit testing

De implementatie van de metriek “Unit testing” is gedaan op basis van de beschrijving van het artikel “A Practical Model for Measuring Maintainability” [[2]](#footnote-2). Hier wordt vermeld dat de beste manier voor het bepalen van deze metriek, m.b.v. een dynamische analyse gedaan moeten worden. Om toch een poging te doen om een metriek te bepalen, is er bij de implementatie gekeken naar het aantal assert statements in de code. Idealerwijs, moet het aantal assert statements groter dan of gelijk zijn aan het aantal paden (dus McCabe waarde) van alle methoden in het systeem. De tabel die genoemd staat onder optie 1 (Unit test coverage) is ook gebruikt in de implementatie om deze metriek op een statische manier te bepalen. De resultaten voor de systemen HyperSQL en SmallSQL zijn de volgende:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| System | Metriek | Rank | System | Metriek | Rank |
| SmallSQL | 0 | -- | HyperSQL | 5 | -- |

Voor beide systemen worden geen assert statements gevonden. Dit is de reden voor de slechts mogelijk score. Meer…

### Conclusie

De SIG mapping voor SmallSQL ziet als volgt uit:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Volume | Complexity per unit | Duplication | Unit size | Unit testing |  |
| ++ | -- | - | - | -- |
| Analysability | x |  | x | x | x | -\0 | |
| Changeablity |  | x | x |  |  | -- | |
| Stability |  |  |  |  | x | -- | |
| Testability |  | x |  | x | x | -- | |

HyperSQL heeft de volgende SIG mapping:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Volume | Complexity per unit | Duplication | Unit size | Unit testing |  |
| + | -- | - | -- | -- |
| Analysability | X |  | x | x | x | - | |
| Changeablity |  | x | x |  |  | -/-- | |
| Stability |  |  |  |  | x | -- | |
| Testability |  | x |  | x | x | -- | |

Conclusie…..

### Bijlage A

Deze bijlage bevat aanvullende informatie betreffende de metrieken die berekend zijn t.b.v. de bepaling van de onderhoudbaarheid van de twee systemen SmallSQL en HyperSQL.

Figuur 1: Duplication SmallSQL

Figuur 2: Duplication HyperSQL

Figuur 3: Complexity SmallSQL

Figuur 4: Complexity HyperSQL

Figuur 5: Unit size SmallSQL

Figuur 6: Unit size SmallSQL

1. Artikel 2 “A practical Model for Measuring Maintainability”, pagina 27 [↑](#footnote-ref-1)
2. Artikel 2 “A practical Model for Measuring Maintainability”, pagina 27 [↑](#footnote-ref-2)