|  |
| --- |
| HUSACCT Tool |
| HU Architecture Compliance Tool  17-04-2012 |
| Iteration Plan Construction Analyse Java |

Team 3

Erik Blanken

Rens Groenveld

Asim Asimijazbutt

Tim Muller

# Inhoudsopgave

1. Belangijkste Mijlpalen 2

2. Belangrijkste Doelstellingen 3

3. Toegekende werkzaamheden (Work Item Asignment) 4

4. Work Item Specification 5

4.1. Functioneel Ontwerp 5

4.4. Test ontwerp & Test constructie 5

5. Evaluatiecriteria 6

# Belangijkste Mijlpalen

Onderstaande lijst is een weergave van de belangrijkste mijlpalen, die in deze constructie-fase te behalen zijn.

|  |  |  |
| --- | --- | --- |
| **Lijst 1. Belangrijkste Mijlpalen van de Constructie-fase** | | |
| **Mijlpaal** | | **Datum (Richtlijn Oplevering)** |
| Start | | 10-04-2012 |
| Aanpassen Architectuur & Architecture Notebook | | 19-04-2012 |
| Aanpassen Proof of concept (Conform nieuwe architectuur) | | 19-04-2012 |
| Aanpassen Functional Design. | | 19-04-2012 |
| Aan kunnen leveren van User Interface | | 23-04-2012 |
| Aanleveren Technical Design | | 26-04-2012 |
| Aan kunnen leveren van onderstaande dependency-types | | |
| 1 | Imports | 20-04-2012 |
| 2 | Attribute-Declarations | 20-04-2012 |
| 3 | Method Invocation & Constructur Invocations | 20-04-2012 |
| 4 | Acces of a property | 23-04-2012 |
| 5 | Exception | 27-04-2012 |
| Onderstaande services correct aanbieden | |  |
| 1 | getRoodModules() | 25-04-2012 |
| 2 | getModulesInModule(module) | 25-04-2012 |
| 3 | getModulesInModule(module, dept) | 25-04-2012 |
| 4 | getDependencies(from) | 26-04-2012 |
| 5 | getDependencies(from, to) | 26-04-2012 |
| Eind | | 27-04-2012 |

# Belangrijkste Doelstellingen

Een overzicht van de doelstelling van deze iteratie zijn te vinden in de onderstaande lijst.

* Architecture implementeren, zodat analyse-componenten van willekeurige taal performant en correct kunnen worden aangeroepen.
* Het Herkennen van de volgende dependency-types:
  + Imports
  + Attribute-Declarations
  + Method Invocation & Constructur Invocations
  + Acces of a property
  + Exceptions
* Het duidelijk kunnen aanbieden van onderstaande services
  + getRoodModules()
  + getModulesInModule(module)
  + getModulesInModule(module, dept)
  + getDependencies(from)
  + getDependencies(from, to)

# Toegekende werkzaamheden (Work Item Asignment)

In de onderstaande lijst is een planning en verdeling te vinden van alle werkzaamheden in deze fase.

|  |  |  |  |
| --- | --- | --- | --- |
| **Lijst 2. Planning & Verdeling van de werkzaamheden in de constructie-fase.** | | | |
| **Naam / Omschrijving** | **Prioriteit** | **Toegekend aan:** | **Geschatte uren** |
| Voortgang & Beheer | | | |
| Iteratie Plan: | 1 | Erik | 2 |
| Git Beheer | 1 | Tim | 6 |
| Aanpassingen Architecture Notebook | | | |
| - Aanpassingen FAMIX-domein-model | 2 | Erik | 1 |
| - Decisions & Justification | 1 | Rens | 1 |
| - Used Design Patterns | 1 | Tim | 2 |
| - Fysiek Klassendiagram | 1 | Rens | 2 |
| - Partitionerings-model | 1 | Erik | 1 |
| Test Ontwerp & Test constructie | | | |
| - Blackbox tests per component | 1 | Tim | 3 |
| - Ontwerp & Onderhoud Stubs | 1 | Tim | 4 |
| - JUnit Tests Onderhoud | 1 | Tim | 2 |
| Technical Report | | | |
| - Technical Report = Proof of concept | 1 | Rens | 7 |
| HUSACCT Systeem | | | |
| - JInternalFrame (conform functional design) | 1 | Asim | 10 |
| - JFrame (Voor -specifieke testing-doeleinden) | 3 | Asim | 5 |
| - Opzetten Architectuur | 1 | Rens & Erik | 15 |
| - JavaGenerator: Attributes | 1 | Rens | 8 |
| - JavaGenerator: Classes | 1 | Erik | 1 |
| - JavaGenerator: Packages | 1 | Tim | 3 |
| - JavaGenerator: Imports | 1 | Erik | 1 |
| - JavaGenerator: Method | 1 | Asim | 3 |
| - JavaGenerator: General Control | 1 | Erik | 7 |
| - JavaGenerator: Exception | 1 | Erik | 5 |
| - Service: getRootModules() | 1 | Asim | 3 |
| - Service: getModulesInModule(from) | 1 | Asim | 4 |
| - Service: getModoulesInModule(from, depth) | 1 | Asim | 3 |
| - Service: getDependencies(from) | 1 | Rens | 4 |
| - Service: getDependencies(from, to) | 1 | Tim | 7 |
| Construction Report | | | |
| Opzetten, Inleveren & Printen Report | 2 | Erik | 2 |
| Presentatie Voorbereiden | 2 | Meeting | 2 |
| Update huidige GIT-versie naar main | 1 | Tim | 2 |

# Work Item Specification

### 4.1. Functioneel Ontwerp

Het functioneel ontwerp moet rekening houden met alle eisen en antwoord geven op de volgende vragen:

1. Which data are needed to do all the types of checks/core functional requirements?
   1. Are they already present in the domain models?
   2. Are they already present in the Famix model?
   3. Including exceptions. Which solutions are possible to handle exceptions?
2. Which data are needed for configuration options.
3. Which data are needed for showing the modular structure (GUI & Graphics)?
4. Which data are needed for reporting the violations (GUI, export)?
5. Which data are needed for showing the violations In the graphical modular structure?

#### 4.2. Architecture Notebook

Het Architecture Notebook moet een aanvulling voor het architecture notebook van het gehele systeem. Ontwerp en beschrijf aanvullingen en verbeteringen die met het systeemdeel van jouw team te maken hebben. Stem af met andere teams die dit raakt en met de beheerder van het architecture notebook.

#### 4.3. Technical Report

Modelmatig bewijs dient te worden gemaakt, om te bewijzen dat de applicatie kan werken volgens de in het architecture-notebook omschreven architectuur.

### 4.4. Test ontwerp & Test constructie

Ontwerp black box testen, gericht op de componentafbakening, dus op de services die de componenten leveren aan elkaar. Construeer de tests. Construeer zonodig stubs (achter de interface klassen).

# Evaluatiecriteria

* Er is overeenstemming over de algehele architectuur van het analyse-component.
* GIT Merge na deze iteratie levert weinig tot geen problemen op en er mogen zich geen build-errors voordoen.
* Services getRoodModules(), getModulesInModule(module), getModulesInModule(module, dept), getDependencies(from), getDependencies(from, to) worden aangeboden en kunnen correct worden afgevangen in andere componenten van de HUSACCT-tool.
* JUnit-tests in het analyse-component leveren positieve resultaten.