Martijn Wils

Plan van aanpak: classification tree tool

[Applicatie 1](#_Toc366665026)

[1 Business requirements 1](#_Toc366665027)

[2 Functional requirements 2](#_Toc366665028)

[3 Design functionaliteiten 3](#_Toc366665029)

[4 Performance 3](#_Toc366665030)

[5 Usability 4](#_Toc366665031)

[6 Optionele functionaliteiten 5](#_Toc366665032)

[7 Scope (sprints scrum) 6](#_Toc366665033)

[7.1 Sprint 1 6](#_Toc366665034)

[7.2 Sprint 2 7](#_Toc366665035)

[7.3 Sprint 3 8](#_Toc366665036)

[7.4 Sprint 4 9](#_Toc366665037)

[7.5 Sprint 5 en 6 10](#_Toc366665038)

[7.6 Overige functionaliteiten 11](#_Toc366665039)

# Versiebeheer

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Datum** | **Auteur** | **Beschrijving** |
| V1.0 | 09/09 | Martijn Wils | Aanmaak document |
| V2.0 | 10/09 | Martijn Wils | Eerste feedback door Piet |
| V2.5 | 11/09 | Martijn Wils | Implementatie van planning ms project |
| V3.0 | 13/09 | Martijn Wils | Tweede feedback door Piet en Benjamin |
| V3.1 | 13/09 | Martijn Wils | Implementeren van MoSCoW principe |

# Prioriteiten

In dit document is de prioriteit van een requirement bepaald via het MoSCoW principe.

* **M** - must haves: deze eisen moeten in het eindresultaat terugkomen, zonder deze eisen is het product niet bruikbaar.
* **S** - should haves: deze eisen zijn zeer gewenst, maar zonder is het product wel bruikbaar.
* **C** - could haves: deze eisen zullen alleen aan bod komen als er tijd genoeg is.
* **W** - won't haves: deze eisen zullen in dit project niet aan bod komen maar kunnen in de toekomst, bij een vervolgproject, interessant zijn.

# Applicatie

Het doel van deze applicatie is om een duidelijke en gratis tool te hebben waarmee classification tree testing mogelijk is. Deze tool zal hoofdzakelijk gebruikt worden voor marketing en opleidingen.

# Business requirements

|  |  |  |
| --- | --- | --- |
|  | M | De applicatie moet simpel te gebruiken zijn, nieuwe gebruikers moeten snel hun weg kunnen vinden binnen de applicatie. |
|  | M | De applicatie moet een positief beeld weergeven over Improve (logo). De applicatie zal namelijk gebruikt worden door de medewerkers, maar ook door de klanten van Improve. |
|  | M | De applicatie moet gemakkelijk te onderhouden zijn na de periode van 29 november (einde van stage). |
|  | M | De applicatie zal ontwikkeld worden in 6 sprints over een tijd van 12 weken. |
|  | M | Na 12 weken moet de applicatie stabiel zijn om te gebruiken voor eigen doelstellingen of voor presentaties. |
|  | M | De applicatie moet in het Engels zijn |

# Functional requirements

|  |  |  |
| --- | --- | --- |
|  | M | Het moet mogelijk zijn om de inhoud van de file op te slaan en terug te openen |
|  | S | De applicatie moet in staat zijn meerdere files te openen: Multi-document interface |
|  | M | De gebruiker van de applicatie moet de mogelijkheid hebben om onderverdelingen te maken op de root van de classification tree. |
|  | M | De applicatie zal ontwikkeld worden in 6 sprints over een tijd van 12 weken. |
|  | M | Na 12 weken moet de applicatie stabiel zijn om te gebruiken voor eigen doelstellingen of voor presentaties. |
|  | M | De applicatie moet in het Engels zijn |
|  | M | Bij het aanmaken van een classification tree(de klasse classification tree) zijn er de volgende mogelijkheden: |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entities | Doel | Toegelaten parents | Toegelaten  Afstammelingen | Test case toepasbaar |
| Root | -denotes the test object |  | \*compositions  \*classifications | x |
| Compositions | -consist-of relationship  -to model an aspect that is broken down into other aspects  -to refine a value consisting of several parts | \*root  \*composition  \*class | \*compositions  \*classifications | x |
| Classifications | -is-a relationship  -to model an aspect that is partitioned into a collectively exhaustive and mutually exclusive set of values (equivalence classes) | \*root  \*composition  \*class | \*class | x |
| Classes | -to model equivalence classes (set of values that elicits the same behavior from the TO) and actual test values | \*classification | \*composition  \*classification | V |

|  |  |  |
| --- | --- | --- |
|  | M | Het moet mogelijk zijn om elke entity een andere naam te geven (eigenschap/properties). |
|  | M | Het moet mogelijk zijn om elke entity een andere beschrijving te geven (eigenschap/properties). |
|  | M | Vertrekkende vanuit de classification tree moet het mogelijk zijn om test cases te laten genereren die voldoen aan de eisen van de gebruiker. De ingebouwde functies voor test cases zijn:   * Minimal combination creates a test suite that uses every class from each classiﬁcation at least once in a test case. * Pairwise combination creates a test suite that uses every class pair from disjunctive classiﬁcations at least once in a test case * Complete combination creates a test suite that uses every possible combination of classes from disjunctive classiﬁcation in a test case |
|  | M | De testcases moeten ook handmatig opgesteld kunnen worden aan de hand van een simpele query. Deze query moet de mogelijkheid geven om alle objecten aan elkaar te koppelen via een \* of een + teken |
|  | M | De tool moet alle testcases die gemaakt worden voor een classification tree overzichtelijk en visueel kunnen weergeven. |
|  | S | Het moet mogelijk zijn om testcases handmatig aan te vinken. |
|  | M | De testcases worden onderverdeeld in verschillende groepen. Bijvoorbeeld al de groepen die gedefinieerd zijn bij ingebouwde functies |
|  | S | Één testcase of een hele groep testcases kan aangevinkt worden als voltooid |
|  | M | De applicatie is voorzien van een werkbalk waar snelle acties mee uitgevoerd kunnen worden. De acties:   * New file * Open file * Save file * Undo last * Redo last * Auto layout * Browse forward to other file * Browse backward to other file |
|  | M | Het moet mogelijk zijn om de testcases te exporteren naar Excel |
|  | C | De tool moet de mogelijkheid hebben om een resultaat op te geven en deze te testen aan de hand van een IF-functie. Deze IF functie moet elke testcase afgaan (de mogelijkheid implementeren om enkele een bepaalde groep van testcases te doen). |
|  | C | De applicatie moet de mogelijkheid geven om requirements in te stellen op classes. Dit aan de hand van een simpele query. Bijvoorbeeld: instellen dat iemand van 14 jaar niet op pensioen kan zijn. Met deze requirements moet rekening gehouden worden wanneer de test cases gegenereerd worden. |
|  | C | De applicatie moet de mogelijkheid geven bij het aanmaken van een file om onmiddellijk alles te definiëren. Dit door middel van een venster waarin ingesteld kan worden: de root, de compositions, de classifications en de classes. |
|  | S | De functie om de CT (classification tree) aan te passen op design vlak |
|  | S | Valid en invalid functionaliteit kunnen instellen op de CT. Dit door middel van visueel aan te geven welke class valid of invalid is |
|  | C | De valid en invalid functionaliteit kan gebruikt worden bij het genereren van testcases. |

# Performance

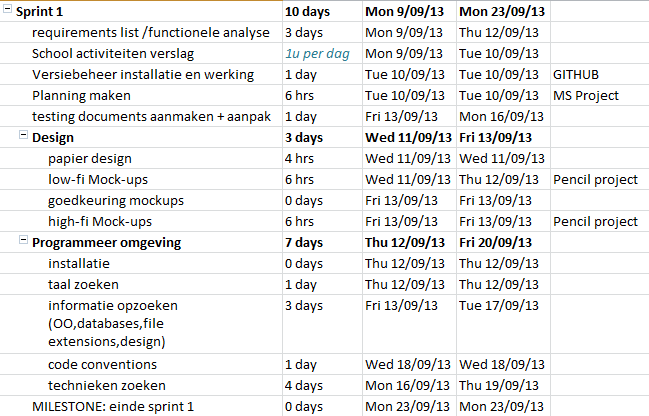
|  |  |  |
| --- | --- | --- |
|  | S | De applicatie moet de file openen in 5 seconden. |
|  | M | De applicatie mag niet crashen bij het uitvoeren van eendere welke actie. |

# Usability

|  |  |  |
| --- | --- | --- |
|  | S | Wanneer de applicatie zijn vorm begint te krijgen en de hoofdfunctionaliteiten aanwezig zijn, zal de applicatie getest worden door 5 of meer personen. Zij zullen enkele basis acties uitvoeren, als deze basis acties binnen aanvaardbare tijd voltooid worden en er geen moeilijkheden ondervonden worden zal de applicatie volstaan aan de eisen van usability. Wanneer er moeilijkheden ondervonden worden zal de lay-out van de appplicatie zodanig worden aangepast zodat het duidelijker wordt. |
|  | S | Na de tests zal de vraag gesteld worden om de UI te beoordelen op lay-out en bruikbaarheid. Beiden met een schaal op 10. De test is geslaagd als er een gemiddeld cijfer van 7 op de schaal van 10 wordt beoordeeld. |
|  | M | De applicatie moet een duidelijk visueel beeld geven van de classification tree met daarop volgend de test cases. |
|  | S | Het moet mogelijk zijn om de classification tree een auto lay-out te geven zodat deze duidelijk en verdeeld weergeven worden. |
|  | S | De applicatie is overkoepelend, hierdoor kunnen er meerdere files geopend worden in hetzelfde programma. |
|  | M | Het moet simpel zijn om elke entiteit te slepen, verplaatsen of verwijderen via de interface. |

# Scope (sprints scrum)

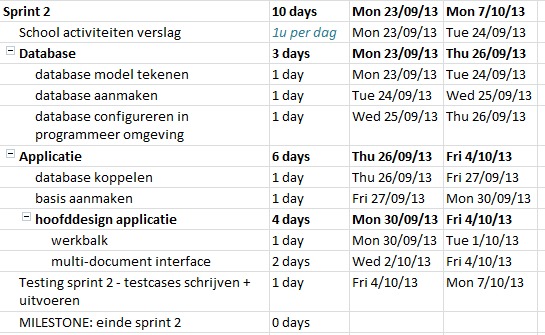
## Sprint 1



Toelichting sprint 1:

Bij deze sprint wordt de basis gelegd voor de volgende weken. De mock-ups zullen hier voorgesteld worden aan de opdrachtgevers en mits hun goedkeuring verder uitgewerkt worden in high fidelity mock-ups die het basis design zullen tonen van de applicatie.   
  
De 2de week van sprint 1 zal vooral in teken staan van het opzoeken en opfrissen van de gekozen programmeertaal/omgeving. Ook zullen de code conventions in deze week bepaald worden, deze zullen nuttig zijn bij het doorgeven van de applicatie.

## Sprint 2



1. Toelichting sprint 2:

Sprint 2 staat in het teken van de database. In deze database zal alle informatie bijgehouden worden over de gemaakte testcases en de classification tree indeling.   
  
Wanneer de database aangemaakt en geïmplementeerd is binnen de programmeeromgeving zal er aan de basis van de applicatie gebouwd worden( de hoofdinterface ).

## Sprint 3



1. Toelichting sprint 3:

Sprint 3 = Classification tree.

In de eerste week van sprint 3 wordt de basis gelegd voor het aanmaken van de CT (classification tree). De klasse zal ontworpen worden en vervolgens geprogrammeerd worden. Wanneer de klasse correct werkt zal er voor gezorgd worden dat de elementen van de klasse visueel aangemaakt kunnen worden in de classification tree file.   
  
Het uitvoeren van de nodige tests is in deze sprint zeer belangrijk aangezien de classification tree de basis is voor het genereren van de testcases. Als dit niet naar toebehoren werkt zal sprint 4 niet mogelijk zijn.

## Sprint 4



1. Toelichting sprint 4

Sprint 4 = testcases gebaseerd op classification tree.

Bij de start van sprint 4 zullen eerst de ingebouwde functies geprogrammeerd worden. Vervolgens de mogelijkheid om de gebruikers ook zelf een functie te laten aanmaken die testcases genereerd.   
  
Wanneer de testcases succesvol aangemaakt worden kan de programmatie beginnen om ze visueel te tonen.

## Sprint 5 en 6



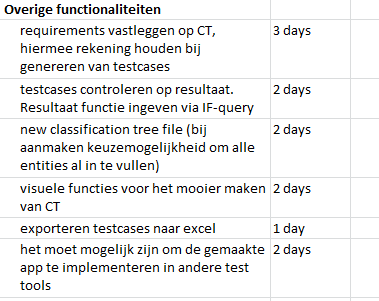
1. Toelichting sprint 5

Wanneer sprint 5 van start gaat zouden alle hoofdfunctionaliteiten van de applicatie aanwezig moeten zijn. Bij sprint 5 is het de bedoeling om de laatste aanpassingen te doen en enkele kleine bugs uit de applicatie te halen. In het slechtste geval wordt sprint 5 gebruikt om functionaliteiten die nog niet volledig klaar zijn te programmeren.

1. Toelichting sprint 6

Sprint 6 is de kleinste sprint, deze duurt namelijk maar 6 dagen en zal met doel hebben de applicatie achterlaten zodat deze later nog uitgebreid kan worden. Ook zal de presentatie over deze applicatie en de documentenstroom voor de PXL in deze sprint verder afgewerkt worden.

## Overige functionaliteiten



1. Toelichting functionaliteiten

Deze functionaliteiten zullen enkel gemaakt worden als er ruimte vrij komt in de planning. Met andere woorden als een functionaliteit sneller af is als gepland. De functionaliteit die gemaakt zal worden als er tijd over is zal op dat moment besproken worden met de opdrachtgevers.