

# Scrum

## Scrum takenlijsten

**Studie:** Technische Informatica, Delft

**School:** Haagse Hogeschool

**Startdatum:** 10 februari 2014

**Einddatum:** 6 juni 2014

**Begeleidende docenten:** Anthony van Geest & Tony Andrioli

**Begeleiders Brightsight:** Rob Bekkers & Remko Foekema

**Stagiair:** Tom Conijn, 10010017



## Inhoudsopgave

1	Inleiding .....	1
2	Implementatie eerste versie Matrix module .....	3
2.1	Takenlijst .....	3
2.2	Taken uitgevoerd per dag .....	5
2.3	Terugblik .....	6
3	Implementatie tweede versie Matrix module .....	7
3.1	Takenlijst .....	7
3.2	Taken uitgevoerd per dag .....	8
3.3	Terugblik .....	10
4	Implementatie gebruikersinterface .....	11
4.1	Takenlijst .....	11
4.2	Taken uitgevoerd per dag .....	12
4.3	Terugblik .....	14



# 1 Inleiding

In dit document zijn de takenlijsten te vinden die zijn gebruikt voor de sprints die zijn uitgevoerd na de oriëntatiefase van drie weken. Er zijn drie hoofdstukken gemaakt waarin een takenlijst te zien is en per dag is opgeschreven welke taken zijn uitgevoerd.

Bij alle sprints is ervoor gekozen om per dag zeven uur beschikbaar te stellen die ingepland kunnen worden. Er is gekozen om zeven uur per dag te kiezen bij een werkdag van acht uur, omdat er altijd tijd verloren gaat tijdens het werken. Dit kan bijvoorbeeld komen door gesprekken met collega's of door een koffiepauze.

Er is voor dit project gekozen om een aangepaste versie van Scrum te gebruiken die beter past bij dit project. Zo wordt er geen gebruik gemaakt van dagelijkse gesprekken die voorgeschreven worden door Scrum, omdat er geen andere projectleden zijn. Dit wordt opgevangen door korte lijnen met de opdrachtgever en de begeleider. Door wekelijks overleg is er controle op de voortgang van het project en worden problemen omtrent de planning snel gedetecteerd.



## 2 Implementatie eerste versie Matrix module

In de eerste sprint is er gewerkt aan het ontwikkelen van de eerste versie van de Matrix module. Deze iteratie gaat vier weken duren, omdat dit is besloten tijdens de keuze voor Scrum. Er worden genoeg taken ingedeeld om deze vier weken in te delen.

### 2.1 Takenlijst

Hieronder is de takenlijst te zien waaruit dagelijks taken gekozen zijn.

Toevoegen benodigde klassen Matrix	7 uur
Weergeven apparaat in Instrument Explorer	1 uur
Toevoegen openen icWaves	5 uur
Toevoegen sluiten icWaves	2 uur
Toevoegen JNA bestanden	2 uur
Vorbereiden bedrijfsbezoek begeleider	2 uur
Uitzoeken hoe excepties gebruikt worden in Java	2 uur
Werking van de script engine onderzoeken	4 uur
Voorbeeldscript proberen in de script engine	2 uur
Implementeren fout afhandeling dll	5 uur
Uitzoeken problemen met openen dll	5 uur
Oplossen problemen met openen dll	1 uur
Uitzoeken welke instellingen nodig zijn voor het doen van metingen	2 uur
Uitzoeken werking property editor	3 uur
Uitzoeken werking enumeraties met de property editor	2 uur
Weergave namen veranderen in de Matrix	3 uur
Uitzoeken hoe een losse meting uitgevoerd moet worden	4 uur
Implementeren losse meting uitvoeren	2 uur

Uitzoeken hoe byte conversie gedaan moet worden	3 uur
Implementeren byte conversie	1 uur
Onderzoeken hoe een oscilloscoop werkt binnen de Matrix	4 uur
Implementeren van oscilloscoop methode	7 uur
Uitzoeken werking Javadoc commentaar	3 uur
Toevoegen Javadoc commentaar	7 uur
Javadoc commentaar verbeteren	3 uur
Uitzoeken werking JUnit	3 uur
Uitzoeken wat er getest moet worden	2 uur
Testen schrijven	7 uur
Eigen implementatie maken van de library	4 uur
Testen verbeteren	4 uur
Verbeteringen uitvoeren naar aanleiding van de testen	3 uur
Vorbereidingen maken voor de code review	7 uur
Versiebeheersysteem kiezen	3 uur
Versiebeheersysteem opzetten	1 uur
Bijwerken UML-diagrammen	5 uur
Werken aan het eindverslag	12 uur
Werken aan het voortgangsverslag	4 uur
Afronden eerste versie module	3 uur



## 2.2 Taken uitgevoerd per dag

Hieronder staat per dag aangegeven welke taken zijn uitgevoerd. Wanneer een taak over verschillende dagen verspreid wordt is elke dag het aantal uren aangegeven dat er aan die taak is gewerkt.

**Maandag 3 maart:** Toevoegen benodigde klassen Matrix.

**Dinsdag 4 maart:** Weergeven apparaat in Instrument Explorer, toevoegen openen icWaves, Toevoegen sluiten icWaves(1 uur).

**Woensdag 5 maart:** Toevoegen sluiten icWaves(1 uur), toevoegen JNA bestanden, uitzoeken werking property editor, werken aan het verslag(1 uur)

**Donderdag 6 maart:** Voorbereiden bedrijfsbezoek begeleider, uitzoeken hoe excepties gebruikt worden in Java, implementeren fout afhandeling dll(3 uur).

**Vrijdag 7 maart:** Implementeren fout afhandeling dll(2 uur) , weergave namen veranderen in de Matrix, uitzoeken welke instellingen nodig zijn voor het doen van metingen

**Maandag 10 maart:** Uitzoeken hoe byte conversie gedaan moet worden, implementeren byte conversie, uitzoeken werking enumeraties met de property editor, uitzoeken problemen met openen dll(1 uur).

**Dinsdag 11 maart:** Uitzoeken problemen met openen dll(4 uur), oplossen problemen met openen dll, werking van de Script Engine onderzoeken(2 uur).

**Woensdag 12 maart:** Versiebeheersysteem kiezen, versiebeheersysteem opzetten, bijwerken UML-diagrammen( 1 uur), werking van de Script Engine onderzoeken(2 uur).

**Donderdag 13 maart:** Voorbeeldscript proberen in de Script Engine, uitzoeken hoe een losse meting uitgevoerd moet worden, implementeren losse meting uitvoeren(1 uur).

**Vrijdag 14 maart:** Implementeren losse meting uitvoeren(1 uur), werken aan het eindverslag(2 uur), onderzoeken hoe een oscilloscoop werkt binnen de Matrix

**Maandag 17 maart:** Implementeren van oscilloscoop methode(5 uur), werken aan het eindverslag(2 uur).

**Dinsdag 18 maart:** Implementeren van oscilloscoop methode(2 uur), uitzoeken werking Javadoc commentaar, toevoegen Javadoc commentaar(2 uur)

**Woensdag 19 maart:** Toevoegen Javadoc commentaar(5 uur), werken aan het eindverslag(2 uur).

**Donderdag 20 maart:** Uitzoeken werking JUnit, uitzoeken wat er getest moet worden, testen schrijven( 2 uur).

**Vrijdag 21 maart:** Testen schrijven(5 uur), eigen implementatie maken van de library(2 uur).

**Maandag 24 maart:** Eigen implementatie maken van de library(2 uur), Javadoc commentaar verbeteren, bijwerken UML-diagrammen(2 uur).

**Dinsdag 25 maart:** Bijwerken UML- diagrammen(2 uur), werken aan het eindverslag(5 uur).

**Woensdag 26 maart:** Testen verbeteren, verbeteringen uitvoeren naar aanleiding van de testen.

**Donderdag 27 maart:** Voorbereidingen maken voor de code review.

**Vrijdag 28 maart:** Werken aan het voortgangsverslag, afronden eerste versie module.

## **2.3 Terugblik**

Tijdens deze sprint was oorspronkelijk het toevoegen van patroonherkenning gepland, maar dit is niet uitgevoerd, omdat hiervoor niet genoeg tijd beschikbaar was. Dit komt vooral doordat het testen en documentatie schrijven meer tijd nodig had dan vooraf verwacht. Ook is er meer tijd in het verslag gaan zitten dan verwacht.

### 3 Implementatie tweede versie Matrix module

In de tweede sprint is er gewerkt aan het ontwikkelen van de tweede versie van de Matrix module. Deze iteratie gaat vier weken duren en ook hier worden taken gekozen, die binnen deze tijd haalbaar zijn.

#### 3.1 Takenlijst

Hieronder is de takenlijst te zien waaruit dagelijks taken gekozen zijn.

Verwerken feedback versie 1	4 uur
Implementeren patrooninstellingen	9 uur
Uitzoeken werking van acties in de Matrix	4 uur
Uitzoeken werking plot-scherm voor selectie	4 uur
Implementeren patroonselectie met het plot-scherm	7 uur
Verbeteren patroonselectie	4 uur
Uitzoeken werking SAD-berekening	2 uur
Implementeren SAD-berekening	6 uur
Uitzoeken hoe de trigger generatie moet werken	2 uur
Implementeren trigger generatie	1 uur
Nieuwe testsituatie creëren	3 uur
Uitzoeken wat de DPA Cardreader doet	2 uur
Uitzoeken hoe de DPA Cardreader aangesloten moet worden	3 uur
Metingen uitvoeren met de testopstelling	10 uur
Patroon kiezen uit gemeten data voor patroonselectie	2 uur
Metingen uitvoeren met geselecteerd patroon	4 uur
Implementeren instelling AC/DC	1 uur

Netbeans upgraden naar versie 8.0	2 uur
Versiebeheer weer werkend krijgen na verplaatsing code	1 uur
Bijwerken UML-diagrammen	3 uur
Javadoc schrijven	3 uur
Tests schrijven	5 uur
Test resultaten verwerken	3 uur
Code coverage tool zoeken	2 uur
Code coverage tools testen	3 uur
Implementeren filter input range selectie	3 uur
Implementeren filter frequentie selectie	4 uur
Javadoc toevoegen voor de filter methodes	3 uur
Testen schrijven voor de filter methodes	3 uur
Werking van de menu actie met behulp van een waveformprovider uitzoeken	7 uur
Uitzoeken werking notify om data te loggen	4 uur
Tweede versie afronden	2 uur
Sprint afronden	1 uur
werken aan het eindverslag	16 uur

### 3.2 Taken uitgevoerd per dag

Hieronder staat per dag aangegeven welke taken zijn uitgevoerd. Wanneer een taak over verschillende dagen verspreid wordt is elke dag het aantal uren aangegeven dat er aan die taak is gewerkt.

**Maandag 31 maart:** Implementeren patrooninstellingen(7 uur).

**Dinsdag 1 april:** Implementeren patrooninstellingen(2 uur), uitzoeken werking SAD-berekening, implementeren SAD-berekening(3 uur).

**Woensdag 2 april:** Implementeren SAD-berekening(3 uur), verwerken feedback versie 1(3

uur).

**Donderdag 3 april:** Verwerken feedback versie 1(1 uur), nieuwe testsituatie creëren, werken aan het eindverslag(4 uur).

**Vrijdag 4 april:** Bijwerken UML-diagrammen, uitzoeken wat de DPA Cardreader doet, uitzoeken hoe de DPA Cardreader aangesloten moet worden(2 uur).

**Maandag 7 april:** Uitzoeken hoe de DPA Cardreader aangesloten moet worden(1 uur), uitzoeken werking van acties in de Matrix, uitzoeken werking plot-scherm voor selectie(2 uur).

**Dinsdag 8 april:** Uitzoeken werking plot-scherm voor selectie(2 uur), implementeren patroonselectie met het plot-scherm(5 uur).

**Woensdag 9 april:** Implementeren patroonselectie met het plot-scherm(2 uur), metingen uitvoeren met de testopstelling(5 uur).

**Donderdag 10 april:** Netbeans upgraden naar versie 8.0, versiebeheer weer werkend krijgen na verplaatsing code, metingen uitvoeren met de testopstelling(4 uur).

**Vrijdag 11 april:** Metingen uitvoeren met de testopstelling(1 uur), patroon kiezen uit gemeten data voor patroonselectie, uitzoeken hoe de trigger generatie moet werken, implementeren trigger generatie, werken aan het eindverslag(1 uur).

**Maandag 14 april:** Verbeteren patroonselectie, werken aan het eindverslag(3 uur).

**Dinsdag 15 april:** Metingen uitvoeren met geselecteerd patroon, implementeren instelling AC/DC, werken aan het eindverslag(2 uur).

**Woensdag 16 april:** Code coverage tool zoeken, code coverage tools testen, javadoc schrijven(2 uur).

**Donderdag 17 april:** Javadoc schrijven(1 uur), tests schrijven, test resultaten verwerken(1 uur).

**Vrijdag 18 april:** Test resultaten verwerken(2 uur), werken aan het eindverslag(2 uur), Implementeren filter input range selectie.

**Dinsdag 22 april:** Implementeren filter frequentie selectie, Testen schrijven voor de filter methodes.

**Woensdag 23 april:** Javadoc toevoegen voor de filter methodes, werken aan het eindverslag(4 uur).

**Donderdag 24 april:** Werking van de menu actie met behulp van een waveformprovider uitzoeken.

**Vrijdag 25 april:** Uitzoeken werking notify om data te loggen, tweede versie afronden, sprint afronden.

### **3.3 Terugblik**

Tijdens deze sprint is de overige functionaliteit toegevoegd, hiervoor was genoeg tijd beschikbaar, omdat er al een goede basis was gelegd tijdens de eerste sprint. Het testen en toevoegen van documentatie heeft minder tijd gekost dan in de eerste sprint, omdat al is uitgezocht hoe dit werkt. Deze sprint was een succes, omdat alle geplande taken zijn gehaald.

## 4 Implementatie gebruikersinterface

In de derde sprint is er gewerkt aan het ontwikkelen van de gebruikersinterface voor de Matrix module. Deze iteratie gaat vier weken duren, omdat dit is besloten tijdens de keuze voor Scrum. Er worden genoeg taken ingedeeld om deze vier weken in te delen.

### 4.1 Takenlijst

Hieronder is de takenlijst te zien waaruit dagelijks taken gekozen zijn.

Uitzoeken werking TopComponents in de Matrix	3 uur
Uitzoeken werking GUI builder in Netbeans	4 uur
Nieuwe repository aanmaken voor de icWaves module	2 uur
Onderzoeken layout andere TopComponents in de Matrix	4 uur
Uitzoeken manieren van weergeven menu's	4 uur
Ontwerpen maken voor labelselectie	4 uur
Implementeren van labelselectie	7 uur
Verbeteren labelselectie	7 uur
Bespreken werking labelselectie	1 uur
Uitzoeken of patroonselectie dynamisch kan	7 uur
JNA bestanden naar een aparte module verplaatsen	1 uur
De locatie van de dll veranderen	2 uur
De dll voor het testen weer werkend krijgen	7 uur
Uitzoeken hoe de lengte van een selectie veranderd kan worden in het plotscherm	2 uur
Implementeren van het aanpassen van de lengte van een selectie	4 uur
Implementeren van controle op maximale patroon lengte	3 uur
Bespreken werking Patroonselectie	1 uur
Bespreken code review	2 uur
Veranderingen doorvoeren naar aanleiding van de code review	4 uur

Uitzoeken hoe het SAD-berekening scherm er uit moet zien	4 uur
Ontwerpen maken van SAD-berekening scherm	5 uur
Gekozen ontwerpen bespreken	2 uur
Naamgeving verbeteren van de menu elementen	3 uur
Uitzoeken/overleggen werking property scherm	3 uur
Implementeren extra opties property scherm	1 uur
Uitzoeken o elementen uit het property scherm gehaald kunnen worden	4 uur
Opstellen systeemtest	7 uur
Uitvoeren systeemtest	2 uur
Toevoegen van testen	4 uur
Javadoc toevoegen/updaten	6 uur
Commentaar toevoegen in de code	4 uur
Werken aan het eindverslag	13 uur
Sprint afronden	3 uur
Code opruimen	4 uur

## 4.2 Taken uitgevoerd per dag

Hieronder staat per dag aangegeven welke taken zijn uitgevoerd. Wanneer een taak over verschillende dagen verspreid wordt is elke dag het aantal uren aangegeven dat er aan die taak is gewerkt. Tijdens deze sprint is maandag 5 mei niet gewerkt, omdat dit een vrije dag is en op dinsdag 13 mei is maar een halve dag gewerkt door andere activiteiten.

**Maandag 28 april:** Uitzoeken werking GUI builder in Netbeans, uitzoeken werking TopComponents in de Matrix.

**Dinsdag 29 april:** Nieuwe repository aanmaken voor hele module, onderzoeken layout andere TopComponents in de Matrix, werken aan het eindverslag(1 uur).

**Woensdag 30 april:** Uitzoeken manieren van weergeven menu's, ontwerpen maken voor labelselectie(3 uur).

**Donderdag 1 mei:** Ontwerpen maken voor labelselectie(1 uur), implementeren van



labelselectie(6 uur)

**Vrijdag 2 mei:** Implementeren van labelselectie(1 uur), bespreken werking labelselectie, verbeteren labelselectie(3 uur), uitzoeken of patroonselectie dynamisch kan(2 uur).

**Dinsdag 6 mei:** Werken aan eindverslag(2 uur), uitzoeken of patroonselectie dynamisch kan (5 uur).

**Woensdag 7 mei:** JNA bestanden naar een aparte module verplaatsen, de locatie van de dll veranderen, werken aan het eindverslag(4 uur).

**Donderdag 8 mei:** Werken aan het eindverslag(5 uur), bespreken code review.

**Vrijdag 9 mei:** De dll voor het testen weer werkend krijgen,

**Maandag 12 mei:** Veranderingen doorvoeren naar aanleiding van de code review, uitzoeken hoe de lengte van een selectie veranderd kan worden in het plot-scherm, implementeren van het aanpassen van de lengte van een selectie(1 uur)

**Dinsdag 13 mei:** Implementeren van het aanpassen van de lengte van een selectie(3 uur), implementeren van controle op maximale patroon lengte(1 uur)

**Woensdag 14 mei:** Implementeren van controle op maximale patroon lengte(2 uur), bespreken werking patroonselectie, verbeteren labelselectie(4 uur)

**Donderdag 15 mei:** Uitzoeken/overleggen werking property scherm, implementeren extra opties property scherm, werken aan het eindverslag (3 uur).

**Vrijdag 16 mei:** Commentaar toevoegen in de code(2 uur), uitzoeken hoe het SAD-berekening scherm er uit moet zien, toevoegen van testen(1 uur)

**Maandag 19 mei:** Toevoegen van testen(3 uur), Ontwerpen maken van SAD-berekening scherm(4 uur)

**Dinsdag 20 mei:** Ontwerpen maken van SAD-berekening scherm(1 uur), naamgeving verbeteren van de menu elementen, gekozen ontwerpen bespreken, uitzoeken of elementen uit het property scherm gehaald kunnen worden(1 uur).

**Woensdag 21 mei:** Uitzoeken of elementen uit het property scherm gehaald kunnen worden(3 uur), Commentaar toevoegen in de code(2 uur), opstellen Systeemtest(2 uur).

**Donderdag 22 mei:** Opstellen systeemtest(5 uur), uitvoeren systeemtest.

**Vrijdag 23 mei:** Sprint afronden, code opruimen.

### **4.3 Terugblik**

Tijdens deze sprint is er gewerkt aan de ontwikkeling van de gebruikersinterface. Hierbij is redelijk goed ingeschat hoeveel tijd er nodig was per onderdeel. Er is ook extra tijd vrijgekomen, omdat na het ontwerpen van het SAD-berekening scherm besloten was deze niet toe te voegen aan de gebruikersinterface. Deze tijd is vervolgens gebruikt om de systeemtest op te stellen en uit te voeren.

Aan het eind van deze sprint moet er alleen nog een handleiding en een presentatie gemaakt worden. Het kan zijn dat in de laatste twee weken van het project nog problemen ontdekt worden en dan moeten deze natuurlijk opgelost worden, maar dit is geen onderdeel van een sprint.