Plan van aanpak



Team: 4

Wouter van den Heuvel 1564952

Wilco Louwerse 1655993

Robin Noten 1671668

Daniel Klomp 1661521

Datum: 12-11-2015

Versie: 0.2

# 

# Inhoud

|  |  |
| --- | --- |
|  |  |
| [Inhoud](#h.9hc0mzvpvt54) | 1 |
| [Inleiding](#h.94qpurv95mxp) | 2 |
| [Onderzoek](#h.1dqh84fyiu8t) | 3 |
| [Op te leveren te producten](#h.a03jpwycmtux) | 4 |
| [Methode van kwaliteitsbewaking](#h.mvyzp1lfjaru) | 5 |
| [Projectorganisatie](#h.13xywu6twofn) | 6 |
| [Projectactiviteiten](#h.q2clffao797b) | 7 |
| [Mijlpalen](#h.7byx79jckasa) | 7 |
| [Teamcontract](#h.julee71refy3) | 7 |
| [Plan van aanpak](#h.mevl1rfv119u) | 7 |
| [Solution architecture](#h.44pk2516gbt2) | 7 |
| [Demonstratie & oplevering eindproducten](#h.ex62ii35xqut) | 7 |
| [Eindbeoordeling](#h.pz1wdrzdqoak) | 7 |
| [Bronvermeldingen](#h.kzpbidsjj2n7) | 8 |
| [Risico’s](#h.2npkxrg8jw85) | 9 |

# Inleiding

- Achtergrond van de opdracht

- Wie heeft om de uitvoering van de opdracht gevraagd (de opdrachtgever)?

- Wat is de aanleiding voor de opdracht? Waarom wil de opdrachtgever dit?

- Wie gaan de resultaten van de opdracht gebruiken (de eindgebruikers)?

- Doel van het document

- Doel van de opdracht (wat aan het einde van het project bereikt moet zijn)

Wij als ‘team 4’ hebben de opdracht gekregen van de Hogeschool Utrecht faculteit Natuur en Techniek binnen de studie Technische Informatica om een webinterface te maken die via een besturingssysteem een wasmachine kan bedienen. Dit project wat dus deel is van de studie Technische Informatica heeft het doel om te leren hoe we een webserver laten communiceren met hardware. Ook leren we hoe een product correct volgens de eisen van de opdrachtgever gemaakt kan worden door middel van een interview en uiteraard het leren om beter samen te werken in teamverband. Binnen deze opdracht vervult een van onze docent de rol als opdrachtgever. De opdrachtgever wilt een nieuwe revolutionaire soort wasmachine op de markt brengen die naast voor de gewone huisvrouw ook bedoeld is voor ondernemingen die speciale of grote wasopdrachten hebben voor bijvoorbeeld een specifiek soort werkkleding.

In dit document zal beschreven worden hoe wij onze doelstellingen voor deze opdracht gaan verwezenlijken. Dit zal gedaan worden door middel van de volgende onderwerpen: … .

Het doel van deze opdracht is zoals eerder al genoemd: het maken van een webinterface die via een besturingssysteem een wasmachine kan bedienen. Er word hierbij gebruik gemaakt van een emulator voor de hardware van de wasmachine en een raspberry pi om commando’s naar deze emulator te kunnen sturen.

# Onderzoek

**In dit hoofdstuk wordt besproken welke informatie moet worden verzameld en op welke manier deze informatie moet worden verkregen. Verder worden de experimenten die moeten worden uitgevoerd nader uitgelegd.**

Om dit product goed in elkaar te kunnen zetten met de juiste functionaliteiten, is het nodig om bepaald informatie te verzamelen. Zo is het bijvoorbeeld belangrijk om te weten wat de functionaliteiten zijn van de wasmachine, maar het is ook om te weten wat de klant belangrijk vindt.

Om de functionaliteiten en de klant behoefden duidelijk te krijgen, wordt een interview gehouden met de klant. Voor dit interview worden van te voren een aantal vragen opgesteld, waaruit duidelijk moet worden wat de klant wilt. Vervolgens worden deze vragen in het vorm van een interview aan de klant gesteld. Op deze vragen wordt verder ingegaan, zodat het een levendig gesprek blijft tussen ontwikkelaar en klant. De uiteindelijk verkregen informatie wordt uiteindelijk in het MoSCoW-model uitgewerkt.

Verder moet het team weten hoe deze eisen moeten wordt geïmplementeerd. De meeste informatie hiervoor wordt door de Hogeschool Utrecht geleverd door middel van een aantal circussen. In deze circussen worden een aantal hoorcollege’s gegeven en moeten een aantal practica worden uitgevoerd.

Wat niet in de cursussen wordt behandel is hoe gewerkt moet worden met Java-script. Om Java-script toch onder de knie te krijgen wordt gebruikt gemaakt van externe bronnen zoals CodeAcadamy en Youtube.

Om dit product te realiseren moeten een aantal experimenten worden uitgevoerd. Zo zal het volgende moeten worden getest:

- De server moet verbinding kunnen maken met de machine.

- De gebruiker moet een wasprogramma kunnen kiezen via de webserver.

- De gebruiker moet een wasprogramma kunnen aanzetten/pauzeren/stoppen.

Dit is allemaal erg makkelijk te testen door de commando’s door de webserver naar de wasmachine te sturen en te observeren of de gewenste resultaten worden laten zien.

Ook moet de beveiliging worden getest. Zo moet de wasmachine niet geopend kunnen worden wanneer een wasprogramma draait. En moet het verwarming’s element altijd uitstaan als er geen water in de machine aanwezig is. Dit wordt getest door de wasmachine aan deze conditie’s bloot te stellen en te controleren of alles naar wens reageert.

# Op te leveren te producten

**In dit hoofdstuk wordt uitgelegd welke producten allemaal moeten worden opgeleverd aan het eind van het project. Deze producten worden uiteindelijk beoordeeld en tellen mee voor het eindcijfer.**

Deze producten zijn:

* Prioritering van op te leveren functionaliteiten
* (Kwaliteits)eisen waaraan de op te leveren producten moeten voldoen
* Een benodigdheden document waarin de functionele en niet-functionele eisen worden vermeld zoals die op basis van een interview met de klant zijn vastgesteld, geprioriteerd met behulp van de MoSCoW-methode.
* Een plan van aanpak.
* Een Requirements Architecture gemaakt met behulp van Software Ideas Modeler versie 8.x.
* Een Solution Architecture gemaakt met behulp van Software Ideas Modeler versie 8.x.
* Een technisch verslag. Het verslag moet voldoen aan de eisen zoals geleerd zijn bij het vak Professional Skills en zoals die beschreven worden in het boek Leren communiceren van Steehouder.
* De HTML-, CSS- en JavaScript-code voor de webpagina’s.
* C++-code voorzien van verduidelijkend commentaar en Doxygen documentatie.
* Een demo van het systeem. Na afloop van de demo moeten vragen over alle aspecten van het systeem kunnen worden beantwoord.

# Methode van kwaliteitsbewaking

**In dit hoofdstuk word besproken hoe wij als team de kwaliteit van onze producten zullen waarborgen.**

Om spel- en taalfouten in onze documentatie tot een minimum te beperken zal telkens wanneer een teamlid een stuk tekst wil aanleveren dit eerst moeten worden doorgelezen (en waar nodig gecorrigeerd) door een ander teamlid.

Om te kunnen garanderen dat onze code goed gedocumenteerd is dient er, telkens wanneer een teamlid code naar de master-branch van de repository wil pushen, door minstens twee teamleden de volgende checklist te worden ingevuld:

* Komt het klassendiagram overeen met de code?
* Is alle code voorzien van commentaar met Doxygen?
* Compileert de code?
* Runt het programma zoals bedoeld?

Pas wanneer er aan deze checklist wordt voldaan mag de code naar de master-branch worden gepusht.

# Projectorganisatie

**Voorzitter** *(Daniel Klomp)*

De belangrijkste taken van de voorzitter zijn om orde te houden in het team door onder anderen te weten en te controleren of de samenwerking en communicatie goed verloopt binnen het team, ook is het van belang dat er goed contact word gehouden met de opdrachtgever door de voorzitter zodat het team goed gestuurd kan worden in de juiste richting dat de opdrachtgever wil dat het project verloopt.

Verder heeft de voorzitter nog een paar andere taken, namelijk het bijhouden van de urenverantwoording en het opstellen van de agenda, de agenda bevat de dagen en tijden waarop er bijeenkomsten of vergaderingen zijn voor het hele team .

**Notulist** *(Robin Noten)*

De notulist heeft de verantwoordelijkheid om tijdens of na vergaderingen de notulen te schrijven, deze moeten de belangrijkste dingen bevatten die er besproken/afgesproken zijn tijdens de vergadering.

**Hardware verantwoordelijke** *(Robin Noten)*

De hardware verantwoordelijke heeft de taak om alle benodigde hardware die de wasmachine simuleert te bewaren en mee te nemen wanneer nodig. Ook zal deze persoon de kwaliteit van de hardware in de gaten houden, dus checken op kleine beschadigingen en dergelijken.

**Raspberry pi verantwoordelijke***(Wilco Louwerse)*

De raspberry pi verantwoordelijke heeft de taak om de raspberry pi en bijbehorende hardware elementen te bewaren en mee te nemen wanneer nodig. Ook zal deze persoon in de gaten houden hoe het zit met de kwaliteit van deze hardware, dit houd in het checken op kleine beschadigingen en defecten.

**Teamlid**

Elk teamlid inclusief de notulist en voorzitter heeft de verantwoordelijkheid om voor zichzelf bij te houden hoeveel uur en waaraan hij per week werkt voor dit project. Dit is zodat de voorzitter tijdens de vergaderingen de urenverantwoording kan bijwerken.

Ook word van elk teamlid verwacht dat hij op tijd aanwezig zal zijn bij gezamenlijke activiteiten met betrekking tot dit project. Indien een teamlid niet aanwezig kan zijn zal hij dit zo snel mogelijk aan de teamleider/voorzitter en de rest van het team vermelden zodat deze daar rekening mee kunnen houden.

# Projectactiviteiten

Voor het voltooien van het project moeten er uiteraard werkzaamheden worden verricht om de eind- en tussenproducten tijdig op te leveren.

## Mijlpalen

Voor het project moeten de volgende mijlpalen behaald, en moeten de bijbehorende documenten of producten als Voldoende worden beoordeeld:

### Teamcontract

Er moet worden vastgelegd hoe we als team samenwerken, en hoe wij met elkaar contact houden. Het document waarin dit wordt gedaan is het Teamcontract, en dit document wordt doorgegeven aan de opdrachtgever om ervoor te zorgen dat de opdrachtgever kennis heeft van de gemaakte afspraken, en eventueel hierop kan inhaken.

### Plan van aanpak

Er moet worden vastgelegd wat de planning is voor het verloop van het project, hoe er met risico’s wordt omgegaan en welke speciale verantwoordelijkheden bij welke teamleden liggen. Hiervoor wordt het Plan van Aanpak opgesteld, en ter verantwoording aan de opdrachtgever opgeleverd.

### Solution architecture

Er moet worden vastgelegd voor welk probleem de opdrachtgever een oplossing zoekt, hoe het eindproduct de eisen van de opdrachtgever vervult, en hoe het eindproduct ontworpen is. Hiervoor wordt de Solution Architecture opgesteld, waarin door middel van UML-schema’s het ontwerp van het eindproduct wordt vastgelegd, en verslag wordt gedaan van de belangrijkste gebeurtenissen tijdens het project.

### Demonstratie & oplevering eindproducten

Aan het einde van het project moet er een demonstratie worden gegeven van de opgeleverde producten en verslagen, mede om te bewijzen dat het opgeleverde eindproduct inderdaad doet wat de opdrachtgever verwacht.

### Eindbeoordeling

Als laatste wordt de eindbeoordeling gegeven. Hierin wordt alles beoordeelt wat het team heeft opgeleverd.

((Wat zou hier moeten komen?))

{placeholder voor urenplanning.xlsx}

# Bronvermeldingen

Om te onderbouwen dat elk teamlid zijn onderzoek heeft gedaan, moet er bij de oplevering van de Requirements Architecture, de Solution Architecture en het Eindverslag vermelding gemaakt worden van de gebruikte bronnen.

# Risico’s

In dit hoofdstuk worden de mogelijke risico’s behandeld. Dit kunnen hardware, software of samenwerking’s problemen zijn. Verder worden de maatregelen genoemd om deze problemen te voorkomen.

Als eerste is kan veel fout gaan bij het gebruik van hardware. Doordat de hardware die gebruikt wordt niet van hoge kwaliteit is, kan het makkelijker kapot gaan. Verder kan, doordat de hardware erg klein is, kleine onderdelen makkelijk kwijtraken.   
Dit kan kan voorkomen worden door voorzichtig om te gaan met de hardware, en duidelijke afspraken te maken binnen het team over het hardware gebruik. Ook is het verstandig om een back-up te maken van de hardware, en in ieder geval voor alle onderdelen een reserve te hebben.

Ook op de samenwerking kan het snel mis gaan. Teamgenoten kunnen hun taken niet uitvoeren of de communicatie kan niet soepel verlopen. Dit zijn problemen die in het begin van het project snel moeten worden opgepakt, omdat ze problemen kunnen veroorzaken voor de rest van het project.   
Dit kan allemaal worden voorkomen door elke week een vergadering te houden, waarin wordt gekeken hoe het gaat in het team en met de teamgenoten individueel. Dit helpt om het probleem zo snel mogelijk het herkennen.

Als laatste kunnen problemen voordoen met de software. Zo kan het medium wat gebruikt wordt om de code te delen defect worden of niet correct worden gebruikt door de teamleden. Hierdoor kan veel werkt dat als is verricht teniet worden gedaan.

Dit kan worden voorkomen door regelmatig lokaal een back-up te maken van de code en de documenten. Verder kan het nog helpen om met de teamleden duidelijke afspraken te maken wanneer code mag worden samengevoerd.