Plan van Aanpak

Wasmachines leven langer met Robijn

Teamnummer: 13

Teamnaam: Robijn

Naam + studentnummer:

1615865 Waila Woe

1660936 Thijs Hendrickx

1665553 Mathijs van Bremen

1657667 Yorick Schellevis

1664653 Bouke Stam

Documentdatum: 3-12-2015

Versienummer: 1.0

# Index

|  |  |
| --- | --- |
| [1. Inleiding](#h.ursy46hijjwv) | 2 |
| [2. Onderzoek](#h.3hexsxifw51o) | 3 |
| [3. Op te leveren producten](#h.49ade1w4sxh1) | 4 |
| [4. Methode van kwaliteitsbewaking](#h.edj3x0kryqz1) | 7 |
| [5. Projectorganisatie](#h.ie9veacp3uyq) | 9 |
| [6. Projectactiviteiten](#h.u4sbtn9nz12b) | 10 |
| [7. Risico’s](#h.8ghpiext8qua) | 12 |
| [8. Bronvermelding](#h.g1ehvljbvpuv) | 14 |
| [9. Bijlagen](#h.2ytwwfjpntyk) | 15 |

# 1. Inleiding

In deze opdracht wordt de kennis van C++ in een wat meer systematische omgeving gebruikt. Het doel is een besturingssysteem te maken voor een wasmachine op het Internet of Things. Hierdoor kan de gebruiker de wasmachine via het internet aansturen op een website, app of programma.

We komen in een tijdperk waarin steeds meer systemen bediend kunnen worden over het internet. Consumenten kunnen lichten, verwarming, televisie en nog veel meer allemaal bedienen met één druk op een knop op hun smartphone. In deze opdracht ligt de focus bij een wasmachine. Van de watertoevoer tot de rotatiesnelheid van de trommel, alles moet door middel van een ingesteld wasprogramma bediend kunnen worden via een makkelijke gebruikersinterface op een website.

Dit verslag dient ervoor om een overzicht te maken van wat het project inhoudt en wat de afspraken tussen het team en de opdrachtgever zijn. Hierdoor krijgt iedereen die bij het project betrokken is een goed beeld van wat de uiteindelijke realisatie zal inhouden en wat er verwacht kan worden van het ontwerp proces.

Het doel is een functionerend gebruikersinterface te hebben dat verscheidene onderdelen van een wasmachine kan bedienen over het internet.

# 2. Onderzoek

Voordat er begonnen kan worden met het bedenken van de oplossing moet er eerst onderzoek worden gedaan naar alle software en hardware die gebruikt gaat worden. Enkele zijn door school aangewezen en van andere moet er worden uitgezocht welke het beste is. Hieronder wordt beschreven wat er voor elk onderdeel moet worden uitgezocht.

**2.1 Raspberry PI**

* Uit welke onderdelen bestaat een raspberry pi?
* Welk operating system gaan we draaien op de raspberry pi?
* De website *Raspberrypi* (z.j.) geeft informatie over de specificaties en operating systems.

**2.2 Webpagina server**

* Wordt er een bestaande server gebruikt of wordt er een gemaakt?

**2.3 Socket server**

* Hoe wordt de server socket gemaakt?

**2.4 JavaScript server communicatie**

* Hoe kan de browser met de socket server communiceren?
* Wat voor berichten moeten allemaal gestuurd worden?
* Welk protocol is handig voor de communicatie?

**2.5 RTOS**

* Welke klassen heeft RTOS?

**2.6 Wasmachine emulator**

* Uit welke componenten bestaat de emulator?
* Hoe verloopt de communicatie tussen de emulator en de raspberry pi?
* In welke toestanden kan de emulator zich bevinden?
* Deze informatie is te vinden in de beschrijving van de wasmachine emulator van Marten Wensink (2015).

**2.7 Wasprogramma**

* Uit welke gegevens bestaat een wasprogramma?
* Hoe kan een wasprogramma worden opgeslagen?

# 3. Op te leveren producten

Gedurende de loop van het project moeten er verschillende producten worden ingeleverd. Deze producten variëren van modellen tot werkend product. Hieronder staat beschreven welke producten er moeten worden ingeleverd en wat er wat de eisen daaraan zijn.

**3.1 Webpagina**

Een webpagina vanwaar de wasmachine bestuurd kan worden via websockets. Deze pagina moet een optie hebben om het ip van de wasmachine in te vullen. Ook moet er een keuze gemaakt kunnen worden uit een lijst met wasprogramma’s. Daarna moet het programma kunnen worden gestart of gestopt. De pagina laat dan zien hoe lang het programma nog duurt.

Naast de standaard wasprogramma’s moet de gebruiker ook handmatig instellingen als temperatuur, rotatie snelheid en tijdsduur kunnen instellen. De webpagina moet via de websocket server geüpdatet worden zodat de informatie constant klopt.

**3.2 Webserver**

Een webserver die de webpagina(s) host. Dit kan een bestaande server als apache zijn of zelf gemaakt. De server moet een http pagina kunnen versturen. Deze server hoeft niet beveiligd te zijn en het is niet vereist dat deze vanaf ip adressen buiten het lokale netwerk toegankelijk hoeft te zijn.

**3.3 Websocket server**

Wordt gebruikt voor de communicatie tussen de browser en het operating system van de wasmachine. De server moet berichten tussen het operating system en de webpagina kunnen sturen en moet een authenticatie mechanisme bevatten. De gebruiker moet een wachtwoord en gebruikersnaam invoeren om toegang te krijgen tot het operating system.

**3.4 Wasmachine operating system**

Het operating system houdt bij welk programma er bezig is en zorgt dat dat gevolgd wordt door de wasmachine. Het operating system geeft hoe lang het programma nog duurt en kan het programma afbreken of starten als de websocket dat bericht heeft gekregen.

Als de stroom uitvalt moet het wasprogramma dat op dat moment bezig was worden hervat waar het afgebroken werd. Als een deel van de wasmachine kapot is moet dat worden laten zien via een melding aan de gebruiker.

**3.5 Requirements architecture**

Use case diagram met een beschrijving van elke use case en de relaties tussen de use cases. Ook een uitleg van de gemaakte keuzes bij het maken hiervan.

**3.6 Solution architecture**

* Activity diagram per must have use case
* Beschrijving van de architectuur van het systeem
* Klassendiagram
* Taakstructurering
* Concurrency model
* STD van de controllers
* Beschrijving van communicatie protocol tussen browser en operating system

**3.7 Technisch verslag**

Een verslag van het technische ontwerp en de realisatie van het product. Een uitgebreide uitleg van de inhoud staat op de website *Sharepoint* (2015).

**3.8 Moscow analyse**

*Must have:*

Operating system

- Wasprogramma’s starten en stoppen

- Wasprogramma’s in stappen uitvoeren

- Wasprogramma’s opslaan in bestanden

- Bij stroomuitval het wasprogramma hervatten wanneer de stroom weer terug is

Websocket server

- Verbinden met de browser

- Login met naam en wachtwoord

- Communicatie protocol

- Berichten ontvangen en versturen

Webserver

- Bestanden hosten

Webpagina

- Verbind met de websocket server via javascript websockets

- IP adres van websocket server handmatig instelbaar

- Start en stop knop

- Wasprogramma uit lijst selecteren

- Handmatig wasprogramma aanpassen

- Status van het draaiende wasprogramma laten zien

- Status constant updaten via websocket

Modellen

- Use case diagram

- Activity

- Klassendiagram

- Taakstructurering

- Concurrency model

- STD van de controllers

- Beschrijving van communicatie protocol tussen browser en operating system

Technisch verslag

- Inleiding

- Onderzoek

- Requirements architecture

- Solution architecture

- Realisatie

- Evaluatie

- Conclusie

- Bronvermelding

*Should have:*

- Websocket server laat alleen verbindingen binnen hetzelfde netwerk toe

- Geluid afspelen als het wasprogramma klaar is

# 4. Methode van kwaliteitsbewaking

Om de kwaliteit van het product te bewaken hebben we als team besloten waar mogelijk producten door experts te laten reviewen. Hiermee zorgen we ervoor dat eventuele punten die binnen het team over het hoofd zijn gezien toch nog boven water komen en verbeterd kunnen worden. Naast deze reviews zullen teamleden zelf ook controles uitvoeren.

**4.1 Webpagina**

* Code moet voldoen aan bepaalde regels
  + Alle ID’s en klassen lowercase en variabelen gescheiden met ‘-’
  + Te lange regels verspreiden over meerdere regels
  + Accolades op zelfde regel
* HTML5, CSS en Javascript moeten voldoen aan de regels die bij de opmaak van deze bepaalde programmeertalen horen. Of de code voldoet aan deze regels kan gevalideerd worden met de “Markup Validation Service” van W3C.
* Functionaliteit van de webpagina(‘s) moet overeenkomen met wat er met de klant is afgesproken. Dit kan gecontroleerd worden door terugkoppeling naar de klant toe.

Om de bovenstaande punten te verwezenlijken zullen teamleden de code van elkaar reviewen en verbeteren waar nodig. Tegen het eind van het project zal een expert op het gebied van programmeren nog een extra controle uitvoeren.

**4.2 Webserver**

* Testen of de webpagina bereikbaar is door middel van het browsen naar de bepaalde pagina via de website.

Bovenstaande test zal door de teamleden zelf uitgevoerd worden.

**4.3 Websocket server**

* Code moet voldoen aan bepaalde regels
  + Functies lowerCamelCase
  + Klassen UpperCamelCase
  + Variabelen lowerCamelCase
  + Accolades op zelfde regel
* Code moet getest worden door teamleden zelf. Mocht het deze tests met succes doorstaan dan kan er nog een test met iemand van buitenaf uitgevoerd worden.

Om de bovenstaande punten te verwezenlijken zullen teamleden de code van elkaar reviewen en verbeteren waar nodig. Tegen het eind van het project zal een expert op het gebied van programmeren nog een extra controle uitvoeren.

**4.4 Wasmachine operating system**

* Code moet voldoen aan bepaalde regels
  + Functies lowerCamelCase
  + Klassen UpperCamelCase
  + Variabelen lowerCamelCase
  + Accolades op zelfde regel
* Code moet getest worden door teamleden zelf. Mocht het deze tests met succes doorstaan dan kan er nog een test met iemand van buitenaf uitgevoerd worden.

Om de bovenstaande punten te verwezenlijken zullen teamleden de code van elkaar reviewen en verbeteren waar nodig. Tegen het eind van het project zal een expert op het gebied van programmeren nog een extra controle uitvoeren.

**4.5 Requirements architecture**

* Use case diagram moet voldoen aan de bijbehorende regels.
* Use case diagram moet worden gereviewd door een expert op dit gebied.

Om de bovenstaande punten te verwezenlijken zal het use case diagram eerst door alle teamleden gecontroleerd worden. Als het die controle doorstaat gaat het diagram naar een expert op dit gebied voor een review.

**4.6 Solution architecture**

* Diagrammen moeten worden gereviewd door één of meerdere experts op dat gebied.

Diagrammen zullen eerst worden gecontroleerd door teamleden. Als het die controle doorstaat zal het gereviewd worden door één of meerdere experts op dat gebied.

**4.7 Technisch verslag**

* Moet voldoen aan de eisen die bij een technisch verslag gelden.
* Spel- en/of taalfouten worden door teamleden gecheckt. Later zal dit nog door iemand van buitenaf gecheckt worden.
* Het technisch verslag moet worden gereviewd door een expert op dit gebied.

Het technisch verslag zal door teamleden gecontroleerd en waar nodig verbeterd worden. Als het de controle van de teamleden doorstaat zal het verslag door een expert gereviewd worden.

# 5. Projectorganisatie

**5.1 De opdrachtnemers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Naam | Extra taak | Email | Telefoonnummer | Woonplaats |
| Bouke Stam | Team leider  Vergadervorzitter(2) | bouke.stam@gmail.com | 0643618009 | Utrecht |
| Yorick Schellevis | vergader voorzitter | y.schellevis@gmail.com | 0624371911 | Almere |
| Mathijs van Bremen |  | mathijs.van.bremen@gmail.com | 0652342107 | Rumpt |
| Thijs Hendrickx | Notulist(2) | thijs.hendrickx@online.nl | 0612815565 | Maarssen |
| Waila Woe | Notulist | waila.woe@student.hu.nl | 0641631887 | De Meern |

**5.2 De opdrachtgever**

De opdrachtgever is Jan Zuurbier van het bedrijf Swirl. Hij is te bereiken voor vragen per mail op [j.zuurbier@hu.nl](mailto:j.zuurbier@hu.nl)

**5.3 Begeleiding vanuit HU**

De begeleider van dit project, Marten Wensink, is te bereiken op m.wensink@hu.nl

# 6. Projectactiviteiten

Hier onder is de lijst met taken per week te vinden. In de bijlage zit een kopie van de volledige planning en uren per taak.

Week 1

* Teamcontract opstellen
* Github aanmaken
* Voortgangsgesprek
* Globale planning maken
* Begin plan van aanpak

Week 2

* Interview vragen bedenken
* Begin requirement document
* Onderzoek
* Plan van aanpak

Week 3

* Interview opdrachtgever
* Requirements architecture af maken
* planning aanpassen aan requirements
* Beginnen aan technisch verslag

Week 4

* Technische verslag
* RTOS informatie verwerken
* Plan van aanpak inleveren

Week 5

* Solution architecture
* Technisch verslag

Week 6

* Solution architecture
* Technisch verslag

Project week 1

* Wasmachine OS maken
* Website maken
* Wasmachine protocol implementeren
* webserver maken/een volledige kiezen

Project week 2

* Website, webserver en OS aan elkaar koppelen
* Technisch verslag inleveren
* Project inleveren

# 7. Risico’s

**7.1 Wijzigingen in project**  
Wijzigingen die gedaan worden tijdens het project zijn vaak een slecht teken. Wanneer dit in een vroege stadium gebeurt zal dat niet zo groot probleem zijn. Maar als dit gebeurd in een wat later stadium dan zou de planning opnieuw gemaakt moeten worden. Sommige taken zijn mogelijk voor niks gemaakt en nieuwe taken komen erbij. Veel tijd gaat hierdoor verloren. Het is dus belangrijk om zo veel mogelijk contact te houden tussen de opdrachtgever en de opdrachtnemers zodat, wanneer wijzigingen ontstaan, het zo vroeg mogelijk worden behandeld.

**7.2 Slechte planning**  
Een slechte planning kan ervoor zorgen dat het project niet op tijd kan worden gerealiseerd. Dit kan bijvoorbeeld komen door een verkeerde inschatting van een onderdeel. Men kan denken dat een onderdeel van een project in korte tijd af kan worden gemaakt. Of juist het tegenovergestelde dat men denkt dat een onderdeel van een project veel tijd gaat kosten. Nog een voorbeeld is dat een onderdeel over het hoofd is gezien en dan het zelfs niet in de planning staat. Het toevoegen van een of meerdere “should haves” is een oplossing. Hierdoor is de planning meer flexibel en kan indien het nodig is een “should have” uit de planning weg worden gehaald.  
  
**7.3 Onvoldoende kennis/onervarenheid**  
Niet iedereen in een team heeft evenveel kennis. De één heeft misschien veel meer kennis dan een ander. Dit kan misschien zijn doordat degene misschien dat al eens eerder heeft gedaan zoals op werk in een bedrijf of juist thuis in vrije tijd als hobby. Bij het verdelen van taken moet hieraan gedacht worden zodat iedereen een taak heeft om aan te werken zonder dat zij/hij te veel tijd verliest met uitzoeken hoe het moet.

**7.4 Onvoldoend motivatie**  
Onvoldoende motivatie kan een groot probleem zijn. Het is belangrijk dat iedereen gemotiveerd is zodat iedereen ook zo goed mogelijk presteert. Wanneer iemand niet gemotiveerd is en weinig bijdraagt aan het project kan dit een mogelijke bedreiging vormen voor het eindresultaat. Het uitzoeken wat hiervan de reden is belangrijk en aan de hand van die reden kan er gekeken worden wat gedaan kan worden. Een oplossing kan zijn:

* Naar training sturen
* Projectlid vervangen door een ander

**7.5 Slechte samenwerking**  
Een goede samenwerking is cruciaal in een project. Mocht het zo zijn dat sommige projectleden niet met elkaar samen kunnen werken dan is het belangrijk om te gaan zoeken naar oplossingen. Het beginnen met het uitzoeken wat hiervan de oorzaak of oorzaken zijn is een must.

**7.6 Ziek**  
Het gebeurd wel eens dat iemand in de projectgroep ziek wordt. Maar mocht het voorkomen dat een project lid of zelfs de gehele projectgroep ziek is, dan moet er overlegd worden bij de opdrachtgever wat hier aan gedaan kan worden en is het misschien dan ook wel nodig om de planning aan de passen.

**7.7 Projectlid niet meer aanwezig**

Wanneer een taakverdeling al is gemaakt en ieder projectlid dus een taak heeft kan het misschien zo zijn dat iemand tijdelijk niet aan project kan werken of zelfs helemaal niet meer. De planning en taakverdeling zou dan moeten worden aangepast of juist helemaal opnieuw worden gemaakt.

# 8. Bronvermelding

Beschrijving wasmachine-emulator (2015). <https://cursussen.sharepoint.hu.nl/fnt/35/TCTI-V2THO6-14/Studiemateriaal/Beschrijving%20wasmachine-emulator.pdf> (geraadpleegd op 3 december 2015)

Inhoud document ‘Technisch Verslag’ themaopdracht Domotica (2015). <https://cursussen.sharepoint.hu.nl/fnt/35/TCTI-V2THO6-14/Studiemateriaal/Inhoud%20Technisch%20Verslag%20themaopdracht%20Domotica.pdf> (geraadpleegd op 3 december 2015)

Raspberry pi model specifications (z.j.). <https://www.raspberrypi.org/documentation/hardware/raspberrypi/models/specs.md> (geraadpleegd op 3 december 2015)

Raspberry pi operating system images downloads (z.j.). <https://www.raspberrypi.org/downloads/> (geraadpleegd op 3 december 2015)

# 9. Bijlagen

**Planning**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Themaopdracht 6** | **Team:** | **13** |  |  |  |  |  |  |  |  |  |  |
|  |  | Bouke [teamleider] | | Thijs | | Mathijs | | Yorick | | Waila | |  |  |
| Weeknr | Omschrijving activiteit | geplande  uren | bestede  uren | geplande  uren | bestede  uren | geplande  uren | bestede  uren | geplande  uren | bestede  uren | geplande  uren | bestede  uren | Totaal  geplande  uren | Totaal  bestede  uren |
| 1 | Teamcontract opstellen | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 5.0 | 5.0 |
| 1 | GitHub aanmaken | 0.2 | 0.2 | 0.3 | 0.3 |  |  |  |  |  |  | 0.5 | 0.5 |
| 1 | Voortgangsgesprek Marten Wensink |  | 0.5 |  | 0.5 |  | 0.5 |  | 0.5 |  | 0.5 | 0.0 | 2.5 |
| 1 | Vergadering 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 5.0 | 5.0 |
| 1 | Globale planning |  |  |  |  |  |  | 1.0 | 1.0 |  |  | 1.0 | 1.0 |
| 1 | Plan van aanpak | 2.0 | 2.0 | 2.0 | 0.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 10.0 | 8.0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Interview opdrachtgever afspraak maken | 1.0 | 1.0 |  |  |  |  |  |  |  |  | 1.0 | 1.0 |
| 2 | Vergadering 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 5.0 | 5.0 |
| 2 | GitHub private maken | 0.5 | 0.5 |  |  |  |  |  |  |  |  | 0.5 | 0.5 |
| 2 | Vragen bedenken voor interview | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 5.0 | 5.0 |
| 2 | Plan van aanpak | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 2.0 | 1.0 | 2.0 | 1.5 | 10.0 | 8.5 |
| 2 | Requirement document |  |  |  |  | 1.0 | 1.0 |  |  |  |  | 0.0 | 1.0 |
| 2 | Vragen voor interview bij elkaar zetten | 1.0 | 1.0 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Interview met opdrachtgever + voorbereiding | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |  |  |
| 3 | Plan van aanpak | 4.0 |  | 4.0 |  | 4.0 |  | 4.0 | 3.5 | 4.0 |  | 20.0 | 3.5 |
| 3 | Uitwerken vragen interview | 1.0 |  | 1.0 |  | 1.0 |  | 1.0 | 1.0 | 1.0 |  |  |  |
| 3 | requirements architecture |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| 3 | Vergadering 3 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 1.0 | 1.0 | 1.0 | 1.0 | 3.5 | 3.5 |
| 4 | **Plan van aanpak inleveren!** |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| 4 | RTOS informatie verwerken |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| 4 | technisch verslag |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| 5 | technisch verslag |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| 5 | Solution architecture |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| 6 | Solution architecture |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| 6 | technisch verslag |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| P1 | Wasmachine OS maken |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| P1 | Website maken |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| P1 | Protocol implementeren |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| P1 | Webserver maken/kiezen |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| P1 | technische verslag |  |  |  |  |  |  |  |  |  |  | 0.0 | 0.0 |
| P2 | Website, server, OS aan elkaar koppelen | 2 |  | 2 |  | 2 |  | 2 |  | 2.0 |  | 10.0 | 0.0 |
| P2 | technisch verslag | 5 |  | 5 |  | 5 |  | 5 |  | 5.0 |  | 25.0 | 0.0 |
| P2 | Bugs in Website, server en OS fixen | 5 |  | 5 |  | 5 |  | 5 |  | 5.0 |  | 25.0 | 0.0 |
| EIND P2 | **Inleveren project** |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **Totaal** | 29.2 | 12.7 | 26.8 | 8.3 | 26.5 | 11.0 | 28.0 | 15.0 | 27.0 | 10.0 | 137.5 | 57.0 |