**Projectplan**

***OS3 Home Alarm System***

*Fontys University of Applied Sciences*

***Eindhoven***

|  |
| --- |
| **Datum : 17-09-2019** |
| **Versie : Versie 2.1** |
| **Status : Versie ter verificieren.** |
| **Auteur : L.J. Hovestadt** |

#### Versie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versie** | **Datum** | **Auteur(s)** | **Wijzigingen** | **Status** |
| 1.0 | 09-09-2019 | L.J. Hovestadt | Initiele versie, gebruikt model gewijzigd na feedback van Michael Franssen | Afgerond |
| 2.0 | 14-09-2019 | L.J. Hovestadt | Initiele versie omgezet naar behoren template | Afgerond |
| 2.1 | 17-09-2019 | L.J. Hovestadt | [1] Paragraaf splitsing versiebeheer [2] Uitbreiding v.d. planning [3] risico inschatting: meer risico’s toegevoegd | Afgerond |

**Verspreiding N.v.t**

|  |  |  |
| --- | --- | --- |
| **Versie** | **Datum** | **Aan** |
|  |  |  |
|  |  |  |

**Inhoudsopgave**

1. Projectopdracht 4

1.1 Context 4

1.2 Doel van het project 4

1.3 Begrenzing 4

1.4 Strategie 4

1.5 Onderzoeksvragen 4

1.6 Eindproducten 4

2. Projectorganisatie 6

2.1 Teamleden 6

2.2 Communicatie 6

3. Activiteiten en tijdplan 7

3.1 Opdeling en aanpak van het project 7

3.2 Overall tijdplan 7

4. Testaanpak en Configuratiemanagement 8

4.1 Testaanpak/strategie 8

4.2 Testomgeving en benodigdheden 8

4.3 Configuratiemanagement 8

5. Financiën en Risico’s 9

5.1 Kostenbudget 9

5.2 Risico’s en uitwijkactiviteiten 9

# Projectopdracht

## Context

*Het project betreft een applicatie die de mogelijkheid biedt om meerdere huizen te beveiligen d.m.v. de Arduino Uno. Op elementair niveau bestaat het systeem uit drie componenten: De arduino plus laptop, de server, en de gebruiker. De gebruiker zal zijn huis kunnen beheren via zijn pc en wordt gemeld via SMS als er een inbraak is. Mogelijke uitbreidingen voor het systeem zijn bijvoorbeeld het nabootsen van een realitische situatie met x aantal huizen, een hoofdbeheerder, en een videocamera en/of microfoon die de inbreker kan identificieren.*

## Doel van het project

Het doel van het project is het realizeren van een relatief complex systeem en het process hiervan. Het project zal ons de studenten introduceren aan wat er bij komt komen in het ontwikkelen van een software project door middel van een formeel software development life cycle model. Qua documentatie zijn de doelen om de verschillende documenten betreft het V-model op te leveren en deze op voldoende niveau te produceren. Naast het documenteren zal de gebouwde applicatie moeten voldoen aan de leerdoelen die opgesteld zijn in semester 3 van de deeltijd opleiding HBO-ICT te Fontys.

Parallel tot de doelen betreft het leren en de persoonlijke ontwikkeling die wij als student zullen ervaren zijn er ook doelen voor het project in zijn eigen geheel. Het doel is om een product op te leveren wat ook daadwerkelijk werkt en op dan wel een functionele wijze gebruikt kan worden.

## Begrenzing

|  |  |
| --- | --- |
| **Tot het project behoort:** | **Tot het project behoort niet:** |
| 1. Communicatie tussen de arduino, server, en de gebruiker met een stateful en stateless verbinding. Meerdere gebruikers die tergelijk de service kunnen gebruiken | 1. Het ontwikkelen van complexe Arduino systemen en het optimalizeren van het fysieke product. Denk hierbij aan i.p.v de arduino te verbinden met de laptop, het direct met de server wordt verbonden. Dit is buiten de scope van ons project. |
| 1. Een desktop applicatie waarop de gebruiker zijn alarm systeem instellingen kan aanpassen. De server API en de relay die met de arduino communiceert. | 1. Een smartphone applicatie. |

## Strategie

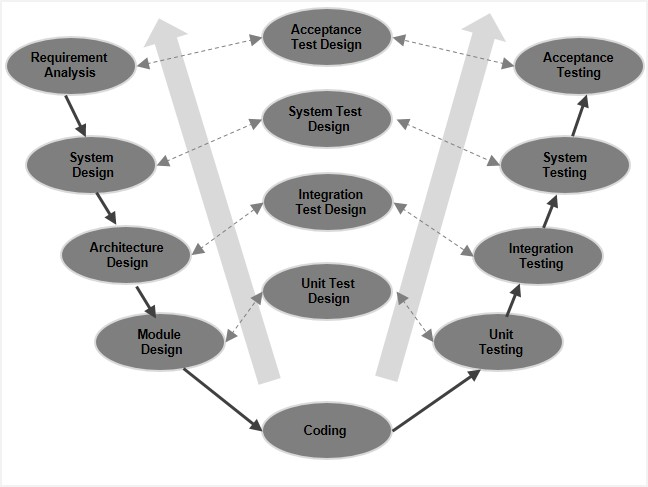
1.4.1 Software Development Cycle Lifespan Model

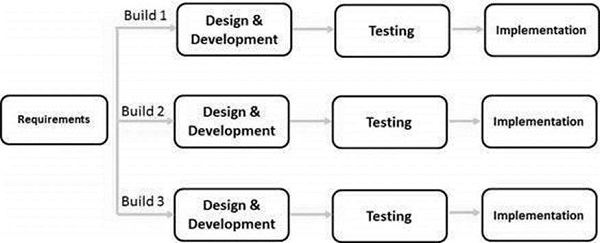
*Het gekozen ontwikkelmodel is het iteratief model. Dit model is gekozen voor een aantal redenen, sommige redenen met een hogere prioriteit dan de ander. Een van de belangrijkste punten is [1] het feit dat er onzekerheid is in de benodigde tijd en dus risiconiveau van de te bouwen features. Door het gebruiken van het iteratief model kunnen de factoren tijd en risico beter beheert worden. [1.1] Een geanticipeerd probleem is dat er te veel tijd in een specifieke feature zou gestopt worden, wij willen dus de mogelijkheid van tijdsnood tegen gaan. Door uit te breiden op vorige versies kunnen we het ons permiteren om uitdagerende requirements op te stellen voor het project. [2] Wijziging van requirements en het toevoegen van features om het project natuurlijk te laten groeien en uitbreiden. [4] De vooruitgang kan makkelijk gemeten worden, elke versie is dus een milestone die bereikt moet worden.*

1.4.2 Wijzigingen betreft het gekozen SDCL model

Na overleg op maandag 9 September met Michael Franssen, onze begeleider van dit project, zijn er een aantal wijzigingen gemaakt in het originele SDCL plan. [1] In der plaats van volledig het iteratief model te gebruiken wordt nu het V-model gebruikt als het primaire SDCL model, i.e., voor de documentatie en testing. [2] De begeleider gaf als argument om het V-model te gebruiken om ons bekwaam te maken met een formeel ontwikkelingsmethodiek. Dit alhoewel onze beredenering solide en onderbouwd in onze keuze was. [3] Het iteratief model wordt gebruikt in de implementatiefase van het V-model. De redenen en onderbouwing voor het gebruik van het iteratief model zijn eerder benoemd. Zie dus ook deze eerder benoemde punten in paragraaf 2.1 hier voor.

Op de volgende pagina staat het diagram van het V-model. Voor een planning en hoeveel tijd wij alloceren per fase zie de projectsplanning.





1.4.3 Versiebeheer

Git is geselecteerd als onze “version control system”, hiermee zullen dus de verschillende versies van de applicatie en elke kleine incrementele verandering gedocumenteerd worden. De primaire reden dat Git is gekozen is door de gemakkelijke toegangbaarheid van de remote repository en het maken van verschillende feature branches.

1.4.4 Werkwijze Git

Elke contributeur zal zijn eigen branch hebben waar code op gecommit wordt. Op deze eigen branches maakt het niet uit als de code wel of niet werkt. Hiernaast zal er een development branch zijn waar de werkende features op moeten staan. Hier komt dus “bug-free” code op te staan, zo ver als dit te realizeren is. De laatste branch zal de master branch zijn. Op de master branch komen de werkende versies van het project. Deze branch kan dus ook als release archief beschouwen worden.

## Onderzoeksvragen N.V.T

## Eindproducten N.V.T

*Reden: productbeschrijving wordt al gegeven in de aanpak van de implementatiefase in de hierna komende paragrafen. Zie Paragraaf 3.1*

# Projectorganisatie

## Teamleden

|  |  |  |  |
| --- | --- | --- | --- |
| **Naam + tel + e-mail** | **Afk.** | **Rol/taken** | **Beschikbaarheid** |
| *Lucas J. Hovestadt (SE) 0647681433*  *Bas de Jong (SE)*  *0643595513*  *Wesley Smits (SE)*  *0611486143*  *Melvin Kusters (SE)*  *634067905* | *Software Engineer (SE)* | *Momenteel zijn er geen specifieke taken. Iedereen zal zijn bijdrage aan het programmeren en documenteren moeten leveren.* | *Personen zijn beschikbaar via onze whatsapp groep. Hierin kunnen mensen zelf bepalen wanneer ze reageren.* |

## Communicatie

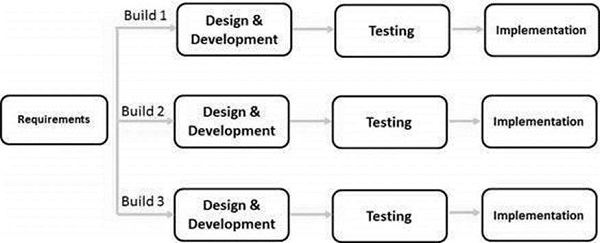
De groep communiceert met de begeleiders via een keer per week een virtuele bijeenkomst te houden. Verder komt de groep bij een op de donderdagen om samen te werken aan de uitdagerendere taken van het project. Op deze donderdagen is er natuurlijk ook een fysiek communicatiekanaal tussen groep en begeleider. De groep communiceert zelf via Whatsapp buiten de lessen om.

# Activiteiten en tijdplan

## Opdeling en aanpak van het project

Zie het hieronder weergeven diagram om de aanpak van de implementatiefase te zien. Let op dat dit een generiek model is en niet een accurate weergave is van hoe wij exact te werk gaan,. Het presenteerd wat het iteratief model is en hoe men in het algemeen dit model toepast. De wijzigingen die wij op dit model maken zijn [1] de documentatie en testing fase beschouwen wij als excessief en zullen niet uitgevoerd worden. De sub-componenten van Design & Development en Testing worden dus uitgevoerd volgens het V-model.Wat blijft is het bouwen van meerdere builds en het maken van de requirements voor dat de implementatie begint. Zie V-model en project planning wanneer de requirements worden gemaakt. Hieronder de verschillende fases die wij momenteel voor ogen zien.

*[1] De eerste fase betreft het ontwikkelen van een minimum viable product, oftewel MVP. In het MVP zullen wij de basis en verticale funtionaliteit bouwen. [2] De tweede fase betreft het verbeteren en het streamlinen van het huidige product. Denk hierbij aan het bouwen van een dashboard om het systeem te monitoren en te beheren. [3] De derde fase zal een van de uitdagerende functies toegevoegd worden zoals bijvoorbeeld het sturen v.d. sms berichten naar de gebruiker. De derde fase kan herhaald worden als er genoeg tijd is om meer complexe features toe te voegen e.g., geluids- en video-opname bij inbraak. [4] Tot slot zal in de laatste fase het systeem geadapteerd worden om in de echte wereld gebruikt kunnen worden. Hiermee zullen we proberen om een echte situatie na te bootsen. Denk hierbij aan meerde huizen die onder bewaking staan van het systeem. Een hoofdbewaker die het complete systeem monitoort, en gebruikers die via een dashboard hun instellingen kunnen aanpassen. Het doel is dus om een narrative en business use te kunnen demonstreren i.p.v. een* geïsoleerd *product.*



## Overall tijdplan

|  |  |  |  |
| --- | --- | --- | --- |
| **Fasering** | **Effort** | **Start** | **Gereed** |
|  |  |  |  |
| 1. Documentatiefase (Volgens het V-model) | 3 weken | Week 2 | Week 5 |
| * 1. Requirement analysis + Acceptatie test design   2. System Design + System Test Design   3. Architecture design + Integration test Design   4. Module design + Unit test design | 1-2 weken  1-2 weken  1-2 weken  1-2 weken | Week 2  Week 2  Week 3  Week 4 | Week 4  Week 4  Week 5  Week 5 (maximaal week 6) |
|  |  |  |  |
| 1. Implementatiefase (Volgens het iteratief model. Iteratie 1: 4 weken, iteratie 2: 2 weken, iteratie 3: 2 weken.) | 8 weken | Week 5 | Week 13 |
| * 1. Iteratie 1:   Eerste versie van het product waarbij meerdere arduino’s een connectie hebben met een main server en hiermee kunnen communiceren en interfaces kunnen aanroepen. (Focus op Must have’s en prototyping.   * 1. Iteratie 2:   Tweede systeem waar de functionaliteit meer gepolished wordt en de should have’s van het product uitgewerkt worden. Enige must have’s die nog niet afgerond waren moeten in deze fase ook af zijn.   * 1. Iteratie 3:   Extra functionaliteiten en de optionele features bouwen. Focus op quality of life type features en het verbeteren van het systeem en gebruiker toegankelijkheid. | Iteratie 1: 4 weken  Iteratie 2: 2 weken  Iteratie 3: 2 weken | Week 5  Week 9  Week 11 | Week 9  Week 11  Week 13 |
|  |  |  |  |
| 1. Testfase | 2 weken | Week 13 | Week 15 |
| * 1. Unit testing   2. Integration testing   3. System Testing   4. Acceptance Testing | 1 week  1 week  1 week  1 week | Week 13  Week 13  Week 14  Week 14 | Week 14  Week 14  Week 15  Week 15 |
| 1. Eindfase | 1 week | Week 16 | Week 17 |
| * 1. Document betreft projectresultaat en evaluatie | 1 week | Week 16 | Week 17 |

# Testaanpak en Configuratiemanagement

## Testaanpak/strategie

*In elke documentatie en ontwerpfase van het V-model wordt een test-plan opgeleverd. In deze documenten komen alle tests te staan die het systeem moet passeren om als compleet beschouwd te worden. De onderbouwing hiervoor is om te verkomen dat je aan de slag gaat met het project en dat ineens een feature van een ander niet werkt. Hiermee zal elke significante commit (dus op de dev branch) al getest zijn. Het werkprocess integreert dus al de filosofie van continu je code te testen.*

## Testomgeving en benodigdheden

*De features zullen door ieder groepslid op zijn eigen branch gemaakt en geverificieerd moeten worden voordat deze op de development branch komt. Bij elke werkende versie zullen we deze van de development branch op de main branch zetten. In principe word het intensieve testen gedaan op de development branch en zal de main branch alleen werkende versies moeten houden. Afhankelijk van welk component getest wordt zullen er verschillende hardware componenten nodig zijn. Het zal mogelijk zijn dat sommige testen alleen werken met alle bijhorende componenten, terwijl een andere feature alleen een laptop nodig heeft.*

## Configuratiemanagement

Het archief zal automatisch worden bijgehouden door git en de verschillende branches die wij gaan gebruiken. Zie paragraaf 1.4.3 voor meer informatie betreft de branches en de opzet voor versiebeheer / archiefmanagement.

# Financiën en Risico’s

## Kostenbudget

*Kosten van de arduinos + multifunctional shield worden uit eigen budget betaald door de individus van het project. Extra benodigdheden zoals gsm module of dergelijke zullen betaald worden door geld te poolen. Wij verwachten geen grote kosten rond om dit project.*

## Risico’s en uitwijkactiviteiten

## Hieronder een diagram betreft de risico’s van het project. Zie de column risiconiveau voor een indicatie van hoe groot de kans is dat dit risico voorkomt. Rangorder van \*..\*\*\*\* waar 4 sterren het hoogste risiconiveau is.

|  |  |  |  |
| --- | --- | --- | --- |
| **Risico** | **Activiteiten ter voorkoming opgenomen in plan** | **Uitwijkactiviteiten** | **Risiconiveau** |
| 1. Defecte apparatuur | Door er voor te zorgen dat iedereen de zelfde aparatuur hebben verdelen we het risico van niet verder te kunnen werken. | Wanner iedereen geen werkende componenten hebben zouden we nieuwe onderdelen moeten aanschaffen. Echter als een iemand zijn component niet werkt zal deze persoon voor zich zelf een nieuw component moeten komen. Toch zal de rest van de groep verder kunnen gaan | \*\*\*  De kans is groot dat er defecte aparatuur zal zijn, echter verwachten wij dat onze preventieve handelingen de consequenties van dit risico sterk verminderen. |
| 1. Te veel tijd besteden aan een feature | Het lesmateriaal goed doornemen om voorbereid te zijn voor de uitdagingen van dit project. | Ondersteuning vragen bij de docenten. | \*\* |
| 1. Ziekte | Hier op valt in principe weinig op te voorkomen behalve het suggesteren van een gezonde levensstijl. Echter het argument voor risico 4 geldt hier ook, al dan wel in lichtere mate. | Als team zullen wij de taken van de persoon die ziek is tijdelijk op moeten nemen | \* of \*\* |
| 1. Wegvallen van een van de groepleden | Doordat wij op harder tempo werken zullen wij een buffer hebben en als een v.d. groepsgenoten wegvalt kunnen we alsnog het project afronden. | Het afronden van het project op gewijzigd tempo. | \* |