|  |  |  |
| --- | --- | --- |
| Opgesteld door | : | Naam schrijver |
| Projectleider | : | Naam projectleider |
| Projectleden | : | Naam projectlid 1  Naam projectlid 2  Naam projectlid 3  Naam projectlid 4  Naam projectlid 5 |
| Begeleider | : | Naam begeleider |
| Datum van uitgifte | : | dd maand jjjj |

Dit document is een template voor een projectdocument van een elektrotechnisch systeem. In de blauwe tekst is uitleg gegeven over de betreffende hoofdstukken. Deze blauwe teksten moet je dus in het uiteindelijke document vervangen door je eigen tekst (en zwart maken).

© 2018 Hogeschool Rotterdam

Projectdocument

van

Systeemnaam

Deze template is algemeen en kan gebruikt worden voor grote en kleine projecten. Het is slechts een template, dit betekent dat een en ander naar eigen inzicht veranderd mag worden zodat het uiteindelijke document beter past bij het specifieke project waar het betrekking op heeft. Dus als paragrafen niet nodig zijn, laat ze dan weg. Als er gegevens zijn die je noodzakelijk vindt, maar waar geen passende paragraaf voor is, maak er dan een paragraaf bij.

Inhoudsopgave

[1 Inleiding 4](#_Toc496046783)

[1.1 Doel 4](#_Toc496046784)

[1.2 Documentconventies 4](#_Toc496046785)

[1.3 Doelgroepen en suggesties voor het lezen 4](#_Toc496046786)

[2 Definitiefase 5](#_Toc496046787)

[2.1 Programma van Eisen 5](#_Toc496046788)

[2.1.1 Algemene beschrijving 5](#_Toc496046789)

[2.1.2 Systeemfuncties 5](#_Toc496046790)

[2.1.3 Niet-functionele klanteisen 6](#_Toc496046791)

[2.2 Acceptatietest 6](#_Toc496046792)

[2.2.1 Voorbereidingen acceptatietest 6](#_Toc496046793)

[2.2.2 Acceptatietestbeschrijvingen 7](#_Toc496046794)

[2.3 Traceerbaarheid klanteisen (Requirements traceability) 8](#_Toc496046795)

[3 Architectuurfase 10](#_Toc496046796)

[3.1 Architectuurontwerp 10](#_Toc496046797)

[3.1.1 Ontwerpbesluiten 10](#_Toc496046798)

[3.1.2 Deelsystemen en koppelingen 10](#_Toc496046799)

[3.1.3 Traceerbaarheid eisen (Requirements traceability) 11](#_Toc496046800)

[3.2 Integratietest 12](#_Toc496046801)

[3.2.1 Aanpak 12](#_Toc496046802)

[3.2.2 Testen 12](#_Toc496046803)

[4 Detailontwerpfase (Detail Design Phase) 13](#_Toc496046804)

[4.1 Detailontwerp 13](#_Toc496046805)

[4.1.1 Unit 1 13](#_Toc496046806)

[4.1.2 Unit 2 13](#_Toc496046807)

[4.2 Unit test 13](#_Toc496046808)

[4.2.1 Unit 1 13](#_Toc496046809)

[4.2.2 Unit 2 14](#_Toc496046810)

[5 Testfase (Test Phase) 15](#_Toc496046811)

[5.1 Unit test resultaten 15](#_Toc496046812)

[5.2 Integratietest resultaten 15](#_Toc496046813)

[5.3 Acceptatietest resultaten 15](#_Toc496046814)

[6 Bibliografie 16](#_Toc496046815)

[Bijlagen 17](#_Toc496046816)

Versiehistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Wijzingen | Auteur |
| 0.1 | dd-mm-jjjj | Beschrijving van de wijzigingen | Naam |
|  |  |  |  |
|  |  |  |  |

# Inleiding

Dit hoofdstuk vormt de inleiding van dit projectdocument. In dit projectdocument wordt ervan uitgegaan dat het V-model wordt gevolgd teneinde een elektrotechnisch systeem te realiseren.

## Doel

Identificeer het elektrotechnisch systeem waarvan het programma van eisen, het ontwerp en de testen in dit document beschreven worden. Beschrijf het doel van het systeem en leg uit welk probleem hiermee opgelost wordt. Deze informatie kun je vinden in het plan van aanpak.

## Documentconventies

Beschrijf standaarden en typografische afspraken die je gevolgd hebt bij het schrijven van dit ontwerpdocument.

## Doelgroepen en suggesties voor het lezen

Beschrijf de verschillende doelgroepen waar dit ontwerpdocument voor bestemd is. Bijvoorbeeld: ontwerpers, project managers, docenten, programmeurs en dergelijke. Geef een korte beschrijving hoe dit projectdocument georganiseerd is. Geef een suggestie hoe dit document het beste gelezen kan worden, bijvoorbeeld door te beginnen met de definitiefase waarna iedere doelgroep naar het hoofdstuk wordt verwezen dat voor hen het meest relevant is.

# Definitiefase

In dit hoofdstuk worden de wensen en ideeën van de klant vastgelegd. Er is vaak uitgebreid overleg met de klant nodig om samen tot klanteisen (Engels: requirements) te komen die haalbaar, eenduidig, specifiek en verifieerbaar zijn. In dit hoofdstuk wordt tevens vastgelegd hoe aan het einde van het project getest kan worden of het systeem aan de klanteisen voldoet. Deze test wordt de acceptatietest genoemd en meestal uitgevoerd in aanwezigheid van de klant.

## Programma van Eisen

De klanteisen van het systeem worden vastgelegd in deze paragraaf. Deze komen dus niet in het plan van aanpak te staan.

De klanteisen worden in het begin van het project opgesteld en dit kan vaak al parallel met het schrijven van het plan van aanpak gedaan worden. De acceptatietest (zie 2.2) wordt gelijktijdig met de klanteisen opgesteld. Dit zorgt ervoor dat alle klanteisen testbaar zijn.

### Algemene beschrijving

Deze paragraaf beschrijft in grote lijnen de omgeving van het systeem en de eisen die aan het systeem gesteld worden.

#### Systeemcontext

Beschrijf de context (omgeving) van het systeem en de relatie met andere systemen. Is dit systeem een nieuwe versie in een bestaande familie van systemen, een vervanging van een bestaand systeem, of een nieuw systeem dat op zichzelf staat? Als dit projectdocument een deelsysteem beschrijft van een groter systeem, relateer de klanteisen van het totale systeem dan aan dit systeem. Een plaatje van de context, zoals een context diagram of een use case diagram kan hierbij helpen.

#### Systeemfuncties

Beschrijf de belangrijkste functionaliteiten (Engels: features) van het systeem. Een functionaliteit wordt ook wel functie genoemd en beschrijft een bepaald gedrag dat het systeem moet vertonen. Details van iedere functie worden in hoofdstuk 2.1.2 gespecificeerd, hier is dus alleen een korte opsomming nodig. Organiseer de functies op een zodanige manier, dat het begrijpelijk is voor de lezer.

### Systeemfuncties

In deze paragraaf worden de belangrijkste functionaliteiten van het systeem beschreven. Bij iedere functie hoort een lijst met klanteisen.

#### Systeemfunctie 1

Verander deze titel in een zinvolle beschrijving voor deze systeemfunctie. Geef een korte globale beschrijving van deze functie. Je kunt eventueel ook de prioriteit van de functie aangeven (hoog, gemiddeld, laag). Je kunt de prioriteit ook aangeven met de zogenoemde MoSCoW methode. Zie eventueel: <https://nl.wikipedia.org/wiki/MoSCoW-methode>.

##### Stimulus en response

Geef de volgorde van gebruikersacties en van de systeemacties weer. Dit geeft inzicht in hoe het systeem moet reageren op input van de gebruiker en vice versa.

##### Functionele en performance klanteisen

Geef hier een opsomming van alle klanteisen die bij deze functie horen. Een functionele klanteis beschrijft wat er gemaakt moet worden, niet hoe dat gedaan moet worden. Het bevat dus geen ontwerpbeschrijving. Klanteisen moeten tenminste identificeerbaar, haalbaar, eenduidig, specifiek en verifieerbaar zijn.

Iedere eis moet dus uniek geïdentificeerd worden met een volgnummer. Bijvoorbeeld:

REQ-1: Beschrijving van klanteis 1 (Engels: requirement 1)

REQ-2: Beschrijving van klanteis 2 (Engels: requirement 2)

REQ-3: Enzovoorts.

#### Systeemfunctie 2

Zie paragraaf 2.1.2.1. Voeg voor elke systeemfunctie weer een nieuwe paragraaf toe. De klanteisen worden gewoon doorgenummerd.

### Niet-functionele klanteisen

Niet-functionele klanteisen specificeren eisen die aan het systeem gesteld worden die niets met de functies van het systeem te maken hebben. Ze beschrijven niet een specifiek gedrag van het systeem maar wel een testbare eigenschap van het systeem. Als de klant bijvoorbeeld eist dat een audioversterker in een rode behuizing wordt opgeleverd, dan is dat een niet-functionele klanteis omdat dit niets met de functie (het versterken van een audiosignaal) van het systeem te maken heeft. Het zou kunnen zijn dat niet-functionele eisen dus wel een “Hoe” beschrijven. Dit is in dit geval geen probleem gezien het feit dat de opdrachtgever het vraagt. Zie ook <https://nl.wikipedia.org/wiki/Requirement>.

## Acceptatietest

In deze paragraaf wordt beschreven hoe aan het einde van het project getest zal worden of het elektrotechnisch systeem aan alle klanteisen voldoet.

### Voorbereidingen acceptatietest

Bij de acceptatietest kan het voorkomen dat er diverse testomgevingen benodigd zijn. Bijvoorbeeld: de eerste serie klanteisen moet getest worden in testomgeving 1, de rest in testomgeving 2. Je krijgt in dat geval meerdere paragrafen waarin de verschillende testomgevingen worden beschreven. In de meeste gevallen zal echter één testomgeving volstaan.

#### Testomgeving 1

Verander deze titel in een zinvolle beschrijving voor deze testomgeving. Geef hier een korte beschrijving van deze testomgeving.

##### Hardware voorbereidingen

Geef aan welke hardware nodig is om de acceptatietest goed uit te kunnen voeren. Je kunt hierbij aan het volgende denken:

* Benodigde hardware en apparatuur: naam en eventueel nummer.
* Benodigde instellingen.
* Kabelaansluitingen en schema van de meetopstelling.
* Stap voor stap instructies hoe de opstelling in gebruik te nemen.

##### Software voorbereidingen

Geef aan welke software er nodig is om de acceptatietest goed uit te kunnen voeren. Je kunt hierbij aan het volgende denken:

* Benodigde software: naam en versienummer.
* Hoe de software te installeren (bijvoorbeeld van memorystick, van netwerk).
* Waar de testsoftware (indien van toepassing) te vinden is (simulatoren, test drivers, databases en dergelijke).
* Stap voor stap instructies hoe de software geïnstalleerd moet worden.

##### Overige voorbereidingen

Geef aan welke voorbereidingen er nog meer nodig zijn, buiten de hierboven beschreven voorbereidingen.

#### Testomgeving 2 (indien van toepassing)

Zie paragraaf 2.2.1.1.

##### Hardware voorbereidingen

Zie paragraaf 2.2.1.1.1.

##### Software voorbereidingen

Zie paragraaf 2.2.1.1.2.

##### Overige voorbereidingen

Zie paragraaf 2.2.1.1.3.

### Acceptatietestbeschrijvingen

In dit hoofdstuk worden alle acceptatietesten gedefinieerd. Een test is gedefinieerd als een bij elkaar behorende groep test cases. Voorbeeld: een van de uit te voeren testen kan bijvoorbeeld het frequentieafhankelijke gedrag zijn. Dit kun je weer onderverdelen in verschillende test cases, bijvoorbeeld eentje voor de maximale versterking bij verschillende frequenties, eentje voor het fasegedrag, eentje voor het testen van de toonregeling enzovoorts.

De test cases zijn gerelateerd aan de klanteisen die beschreven zijn in het programma van eisen. Het is mogelijk om in één test case meerdere klanteisen te testen, dit hoeft dus geen één op één relatie te zijn. Meestal zal er echter per eis een acceptatietest case zijn beschreven.

#### Test 1

Verander deze titel in een zinvolle beschrijving voor deze test. Geef hier een korte beschrijving van het doel van deze test.

##### Test case 1.1

Verander deze titel in een zinvolle beschrijving voor deze test case. Geef hier een korte beschrijving van het doel van deze test case. Completeer vervolgens tabel 1.

Tabel . Beschrijving van test case 1.1.

|  |  |
| --- | --- |
| Klanteis(en): | Geef het nummer of de nummers aan van de eis(en) (bijvoorbeeld REQ-1) die hier getest wordt of worden. |
| Voorwaarden: | Aan welke voorwaarden of condities moet voldaan zijn alvorens deze test case uitgevoerd kan worden? Welke testomgeving, zie hoofdstuk 2.2.1.1, wordt hier gebruikt? |
| Testinvoer: | Geef een beschrijving van de benodigde testinvoer. Denk bijvoorbeeld aan:   * Instellingen van bijvoorbeeld een frequentiegenerator. * Volgorde van invoer van verschillende waarden. |
| Testuitvoer: | Beschrijf hier het resultaat van de test (de testuitvoer). Hetgeen gemeten of bekeken moet worden. Denk bijvoorbeeld aan:   * Gewenste nauwkeurigheid van gemeten waarden. * Instellingen van bijvoorbeeld een multimeter. * Volgorde van meten van verschillende waarden. |
| Criteria: | Beschrijf hier waar het testresultaat aan moet voldoen om het resultaat van deze test case op ‘goed’ te krijgen. Specificeer hier dus zaken als onnauwkeurigheid, toleranties, aantal keren dat overnieuw getest mag/moet worden en dergelijke. Als het testresultaat niet aan de criteria voldoet, wordt de test case met ‘fout’ beoordeeld. |
| Testprocedure: | Geef hier een stap-voor-stap beschrijving hoe de test case uitgevoerd moet worden. Deze beschrijving moet zodanig zijn dat een andere tester zonder verdere hulp deze test op de juiste manier kan uitvoeren. |

##### Test case 1.2

Zie paragraaf 2.2.2.1.1. Voeg voor elke benodigde test case weer een nieuwe paragraaf toe.

#### Test 2

Zie paragraaf 2.2.2.1. Voeg voor elke benodigde test weer een nieuwe paragraaf toe met subparagrafen voor de verschillende test cases.

## Traceerbaarheid klanteisen (Requirements traceability)

Requirements traceability is een methode om aan te geven hoe de verschillende klanteneisen (beschreven in paragraaf 2.1) gerelateerd worden aan de verschillende acceptatietesten (beschreven in paragraaf 2.2).

Alle klanteneisen moeten gerelateerd kunnen worden aan een acceptatietest case, want anders zou er een eis niet getest worden. Een traceerbaarheidstabel (traceability table) is een handig hulpmiddel om te kijken of er geen test vergeten wordt.

Een traceerbaarheidstabel die in definitiefase wordt gemaakt ziet er uit zoals tabel 2.

Tabel 2. Requirements per test case.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | REQ-1 | REQ-2 | REQ-3 | Enzovoort |
| Test case 1.1 | X |  | X |  |
| Test case 1.2 |  | X |  |  |
| Test case 2.1 |  | X |  |  |
| Enzovoort |  |  |  |  |

In het hier als voorbeeld gegeven geval worden klanteisen REQ-1 en REQ-3 dus getest in test case 1.1 en wordt klanteis REQ-2 in totaal door twee test cases gezamenlijk getest.

# Architectuurfase

In dit hoofdstuk wordt de architectuur van zowel de hardware als de software van het te ontwerpen elektrotechnisch systeem vastgelegd. In dit hoofdstuk wordt tevens vastgelegd hoe, nadat alle losse units zijn getest, de losse units geïntegreerd kunnen worden tot het beoogde systeem en welke testen daarbij worden uitgevoerd. Deze testen worden integratietesten genoemd.

## Architectuurontwerp

Het architectuurontwerp beschrijft het systeem in grote lijnen.

### Ontwerpbesluiten

Bij het analyseren van de klanteisen worden er al besluiten genomen hoe het ontwerp er in hoofdlijnen uit gaat zien. De belangrijkste functionaliteiten zijn al geïdentificeerd.

In deze paragraaf kunnen deze alternatieven en besluiten gedocumenteerd worden. Voorbeelden van deze ontwerpbesluiten kunnen zijn:

* Ontwerpbesluiten met betrekking tot invoer die wel of niet geaccepteerd wordt.
* Ontwerpbesluiten die aangeven hoe gereageerd moet worden op bepaalde soorten invoer, indien dit niet op voorhand duidelijk wordt uit de klanteisen.
* Een beschrijving die aangeeft hoe de niet-functionele klanteisen (veiligheid, beveiliging, kwaliteit en dergelijke) aangepakt gaan worden.

### Deelsystemen en koppelingen

Deze paragraaf geeft een overzicht van:

* De belangrijkste deelsystemen waaruit het systeem bestaat.
* De koppelingen tussen deze deelsystemen. Een koppeling bevat aan beide zijden een zogenaamde interface.

Bij iedere opsplitsing van een systeem in deelsystemen ontstaan er koppelingen tussen deze deelsystemen.

Het is verstandig hier de functionele decompositie uit te voeren. De benodigde functionaliteiten zijn bekend maar hoe zijn deze aan elkaar gerelateerd? Teken hier bijvoorbeeld een data flow diagram (DFD) . Vanuit dit diagram kun je een blokschema maken. Welke blokken implementeren welke functies?

Er zijn dus twee schema’s; De functies die het systeem heeft (DFD) en het blokschema met blokken die deze functies gaan uitvoeren. Het kan bijvoorbeeld zo zijn dat een CPU veel functies implementeert uit het functionele diagram.

Het blokschema (ook wel AID = Architecture Interconnect Diagram genoemd) is een plaatje op waarin de verschillende deelsystemen te zien zijn, samen met de koppelingen tussen die deelsystemen.

#### Beschrijving van de deelsystemen

In deze paragraaf worden de verschillende deelsystemen een voor een beschreven. De functie van het deelsysteem wordt hier beschreven (wat het deelsysteem doet), de implementatie (hoe het deelsysteem zijn functie uitvoert) wordt pas later beschreven in het gedetailleerde ontwerp, zie paragraaf 4.1. Geef hier ook de specificaties van de deelsystemen. Specificaties zijn eisen die je zelf opstelt. De specificaties worden later bij de integratie- en unittest getest.

##### Deelsysteem 1

Verander deze titel in een zinvolle beschrijving voor dit deelsysteem. Geef een korte globale beschrijving van de functie van dit deelsysteem (algoritme, formule of iets dergelijks) en hoe de inputs en de outputs van het deelsysteem aan elkaar gerelateerd worden (specificaties!).

Bijvoorbeeld:

SPEC-1.1: Output Y geeft de waarde van de temperatuur in gehele graden Kelvin met een afwijking van maximaal 1%.  
SPEC-1.2: De berekening …. moet tenminste 2 keer per seconden uitgevoerd worden.

Bij grote systemen kan een deelsysteem vaak weer opgedeeld worden in meerdere deelsystemen. In dat geval worden deze deelsystemen in aparte sub paragrafen van deze paragraaf beschreven. Een deelsysteem dat niet verder opgesplitst wordt in deelsystemen wordt een unit genoemd.

Indien nodig wordt deelsysteem 1 opgesplitst in meer deelsystemen totdat er bij units terecht wordt gekomen.

##### Deelsysteem 2

Zie paragraaf 3.1.2.1.1. De specificaties van dit deelsysteem worden geïdentificeerd met SPEC-2.1, SPEC-2.2, enz. Dit zorgt ervoor dat de specificaties van de deelsystemen onafhankelijk van elkaar opgesteld kunnen worden. Voeg voor elk deelsysteem weer een nieuwe paragraaf toe.

#### Beschrijving van de koppelingen

In deze paragraaf worden de verschillende koppelingen tussen de deelsystemen een voor een beschreven. Ook hier zijn specificaties op te stellen. Je kunt bijvoorbeeld eisen dat tussen deelsysteem 1 en deelsysteem 2 een ethernet verbinding aanwezig is met een communicatiesnelheid van minimaal 10 Mb/s.

##### Koppeling 1 tussen deelsysteem 1 en deelsysteem 2

Vul in de titel de namen van de betreffende deelsystemen in. Geef een duidelijke beschrijving van de werking van deze koppeling.

Geef hier ook de specificaties van de betreffende koppeling. Bijvoorbeeld:

I(Interconnect)Spec-1.1: Deelsysteem 1 communiceert met deelsysteem 2 via een ethernet verbinding met een communicatiesnelheid van minimaal 10 Mb/s.

##### Koppeling 2 tussen deelsysteem 1 en deelsysteem 3

Zie paragraaf 3.1.2.2.1. De specificaties van deze koppeling worden geïdentificeerd met ISPEC-2.1, ISPEC-2.2, enz. Dit zorgt ervoor dat de specificaties van de koppelingen onafhankelijk van elkaar opgesteld kunnen worden. Voeg voor elke koppeling weer een nieuwe paragraaf toe.

### Traceerbaarheid klanteisen (Requirements traceability)

Requirements traceability is een methode om aan te geven hoe de verschillende klanteisen (zoals beschreven in het programma van eisen) gerelateerd worden aan de verschillende units.

Alle eisen moeten gerelateerd kunnen worden aan een unit, want anders zou er een eis niet geïmplementeerd worden. Een traceerbaarheidstabel (traceability table) is een handig hulpmiddel om te kijken of er geen requirement vergeten wordt.

Een traceerbaarheidstabel die in de architectuurfase wordt gemaakt ziet eruit als tabel 3.

Tabel . Requirements per unit.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | REQ-1 | REQ-2 | REQ-3 | Enzovoort |
| Unit 1 | X |  | X |  |
| Unit 2 |  | X |  |  |
| Unit 3 |  | X |  |  |
| Enzovoort |  |  |  |  |

In het hier als voorbeeld gegeven geval wordt klanteneis REQ-1 dus geïmplementeerd door unit 1, wordt de eis REQ-2 door twee units gezamenlijk geïmplementeerd en wordt eis REQ-3 door unit 1 geïmplementeerd.

## Integratietest

In deze paragraaf wordt beschreven hoe de verschillende deelsystemen, zoals gedefinieerd in paragraaf 3.1, geïntegreerd worden tot het volledige systeem en welke testen daarbij worden uitgevoerd.

### Aanpak

Beschrijf in deze paragraaf in welke volgorde je de incrementele testen aan gaat pakken. Dit beschrijft tevens de manier waarop je alle gemaakte deelsystemen gaat integreren tot een systeem. De integratietestbeschrijving dient als instructie zowel voor de integratie als voor de daadwerkelijke testen die je hierbij uit moet voeren.

### Testen

Geef hier een korte introductie betreffende de integratie en de daarbij uitgevoerde testen.

#### Test 1

Verander deze titel in een zinvolle beschrijving voor deze test. Geef een beschrijving van de integratiestap die hier uitgevoerd en getest wordt en vul tabel 4 in.

Tabel . Beschrijving van integratietest 1.

|  |  |
| --- | --- |
| Voorgaande test: | Verwijzing naar de voorgaande test. |
| Volgende test: | Verwijzing naar de volgende test. |
| Geteste deelsystemen: | Welke deelsystemen worden nu aan elkaar gekoppeld? Bijvoorbeeld: deelsysteem 1 en 2. |
| Testvoorbereiding: | Testomgevingen en dergelijke. |
| Specificaties: | Welke specificatie(s) worden door deze test aangetoond? Bijvoorbeeld: ISPEC-1.1. |
| Testinvoer: | Wat wordt als input gegeven aan welk deelsysteem? |
| Testuitvoer: | Wat verwacht je aan de uitgang van welk deelsysteem? |
| Criteria: | Beschrijf hier waar het testresultaat aan moet voldoen om door te mogen gaan naar de volgende stap in de integratie. Specificeer hier dus zaken als onnauwkeurigheid, toleranties, aantal keren dat overnieuw getest mag/moet worden en dergelijke. Bijvoorbeeld: na 3 keer testen met een maximale afwijking van 3%. Als het testresultaat niet aan de criteria voldoet, wordt de test case met ‘fout’ beoordeeld. |
| Testprocedure: | Geef hier een stap-voor-stap beschrijving hoe de test case uitgevoerd moet worden. Deze beschrijving moet zodanig zijn dat een andere tester zonder verdere hulp deze test op de juiste manier kan uitvoeren. |

#### Test 2

Zie paragraaf 3.2.2.1. Voeg voor elke test weer een nieuwe paragraaf toe.

# Detailontwerpfase

In dit hoofdstuk wordt het gedetailleerd ontwerp van de in paragraaf 3.1.2 gedefinieerde deelsystemen beschreven. Daarnaast wordt ook beschreven hoe deze deelsystemen (units) worden getest. Deze tests worden unit tests genoemd.

## Detailontwerp

Nadat de eisen die aan het systeem worden gesteld zijn vastgelegd in het programma van eisen, en de specificaties van de units (deelsystemen die niet verder opgesplitst zijn) zijn vastgelegd in het architectuurontwerp (paragraaf 3.1), kan begonnen worden met het ontwerpen van de units. De specificaties definiëren wat elke unit moet doen, bij het ontwerpen wordt bepaald hoe deze unit dit gaat doen. Het is de bedoeling dat het ontwerp van elke unit zover gedetailleerd wordt, dat de bouwer de unit ook daadwerkelijk kan bouwen (de schakeling realiseren en/of de software schrijven).

### Unit 1

Verander deze titel in een zinvolle beschrijving voor deze unit. Op basis van de globale beschrijving in paragraaf 3.1.2.1.1 wordt het ontwerp van unit 1 hier verder uitgewerkt. De uitwerking dient zodanig te zijn dat op basis van deze uitwerking de unit geïmplementeerd kan worden.

Voor een hardware unit kun je in deze paragraaf bijvoorbeeld het volgende plaatsen:

* Een beschrijving van de implementatie van de unit. Vergeet niet om te verwijzen naar eventuele bronnen en om de gemaakte keuzes te onderbouwen. De implementatie van een unit wordt vaak gebaseerd op een referentieontwerp uit een datasheet of op een voorbeeld uit een boek of van het internet.
* Een schema van de unit en uitleg over de werking ervan.
* Berekeningen aan componenten en dergelijke.

Voor een unit die software bevat of alleen uit software bestaat kun je in deze paragraaf bijvoorbeeld het volgende plaatsen:

* Een beschrijving en uitleg van de gebruikte algoritmen. Vergeet niet om te verwijzen naar eventuele bronnen.
* Een toestandsdiagram die het gedrag van de unit beschrijft.
* Een structure chart met de aanroepvolgorde van de diverse functies waaruit de unit bestaat.
* Een flowchart die de implementatie van een functie of unit beschrijft.
* Een beschrijving van de belangrijkste functies waaruit de unit bestaat met hun invoer en uitvoer.

### Unit 2

Zie paragraaf 4.1.1. Voeg voor elke unit weer een nieuwe paragraaf toe.

## Unit test

In deze paragraaf wordt beschreven hoe de units, waarvan het gedetailleerde ontwerp in paragraaf 4.1 beschreven is, worden getest.

### Unit 1

Vervang deze titel door de naam van de te testen unit.

#### Test case 1.1

Verander deze titel in een zinvolle beschrijving voor deze test case. Geef hier een korte beschrijving van het doel van deze test case. Completeer vervolgens tabel 5.

Tabel . Beschrijving unittest case 1.1.

|  |  |
| --- | --- |
| Specificaties: | Geef het nummer of de nummers aan van de specificatie(s) (bijvoorbeeld SPEC-1.1) die hier getest wordt of worden. Dit zijn eisen die voortkomen uit het architectuur ontwerp of uit de gevolgen van het detail ontwerp. |
| Voorwaarden: | Aan welke voorwaarden of condities moet voldaan zijn alvorens deze test case uitgevoerd kan worden? Welke testomgeving, zie hoofdstuk 2.2.1.1, wordt hier gebruikt? Als er software nodig is, beschrijf deze ook in de testomgeving, inclusief (een verwijzing naar) de benodigde code! |
| Testinvoer: | Geef een beschrijving van de benodigde testinvoer. Denk bijvoorbeeld aan:   * Instellingen van bijvoorbeeld een frequentiegenerator. * Volgorde van invoer van verschillende waarden. * Wat voor data wordt verwacht? Bijvoorbeeld een registerwaarde over de I2C bus. |
| Testuitvoer: | Beschrijf hier het resultaat van de test (de testuitvoer). Hetgeen gemeten of bekeken moet worden. Denk bijvoorbeeld aan:   * Gewenste nauwkeurigheid van gemeten waarden. * Instellingen van bijvoorbeeld een multimeter. * Volgorde van meten van verschillende waarden. * Bijvoorbeeld I2C data |
| Criteria: | Beschrijf hier waar het testresultaat aan moet voldoen om het resultaat van deze test case op ‘goed’ te krijgen. Specificeer hier dus zaken als onnauwkeurigheid, toleranties, aantal keren dat overnieuw getest mag/moet worden en dergelijke. Als het testresultaat niet aan de criteria voldoet, wordt de test case met ‘fout’ beoordeeld. |
| Testprocedure: | Geef hier een stap-voor-stap beschrijving hoe de test case uitgevoerd moet worden. Deze beschrijving moet zodanig zijn dat een andere tester zonder verdere hulp deze test op de juiste manier kan uitvoeren. |

#### Test case 1.2

Zie paragraaf 4.2.1.1. Voeg voor elke benodigde test case weer een nieuwe paragraaf toe.

### Unit 2

Zie paragraaf 4.2.1. Voeg voor elke unit weer een nieuwe paragraaf toe met subparagrafen voor de verschillende test cases.

# Testfase (Test Phase)

Na de realisatiefase worden, bij het werken volgens het V-model, achtereenvolgens de unit test (beschreven in paragraaf 4.2), integratietest (beschreven in paragraaf 3.2) en acceptatietest (beschreven in 2.2) uitgevoerd. De resultaten van al deze tests worden in dit hoofdstuk beschreven.

## Unittestresultaten

Na het maken en realiseren van de units, worden de diverse unittests (zie paragraaf 4.2) uitgevoerd. In deze paragraaf komt per test het resultaat te staan met bewijsvoering (foto’s, screenshots, meetwaarden enzovoort). Mocht een unit niet voldoen aan de specificaties dan moet worden teruggegaan naar de detailontwerpfase om te kijken of hier fouten zijn gemaakt. Deze fouten moeten worden verbeterd en de V van het V-model moet opnieuw gevolgd worden.

## Integratietestresultaten

Nadat alle units naar behoren functioneren kan de integratie met bijbehorende integratietest worden uitgevoerd. De integratie vindt dus plaats zoals beschreven in de integratietest (zie paragraaf 3.2). Ook hier worden resultaten onderbouwd met bewijsvoering. Mochten hier specificaties niet worden gehaald, dan moet worden teruggegaan naar de architectuurfase.

## Acceptatietestresultaten

Wanneer de integratie goed is verlopen kan een afspraak met de klant worden gemaakt voor de acceptatietest (zie paragraaf 2.2). Deze test alle klanteisen en bewijst aan de opdrachtgever dat het gemaakte systeem voldoet aan alle klanteisen.

# Bibliografie

**There are no sources in the current document.**

Geef een opsomming van alle documenten en referenties waar naar verwezen wordt. Denk hierbij aan zaken als standaarden en het plan van aanpak. Maak bij bronvermeldingen gebruik van de IEEE-stijl.

# Bijlagen

Veel gebruikte bijlagen bij de definitiefase zijn:

* Glossary: alle termen en afkortingen die noodzakelijk zijn om de requirements te kunnen begrijpen.
* Analysemodellen: als er gebruik is gemaakt van een aparte analysemethode bij het opstellen van de eisen of specificaties, dan kan de output daarvan hier opgenomen worden (context diagram, data flow diagrammen, toestandsdiagrammen, use case diagrammen en dergelijke). Tijdens de opleiding Elektrotechniek wordt een bepaalde analysemethode aangeleerd.
* Issues List: een (dynamische) lijst van alle openstaande onduidelijkheden met betrekking tot de requirements.

Veel gebruikte bijlagen bij de ontwerpfase zijn:

* Glossary: alle termen en afkortingen die noodzakelijk zijn om het ontwerp te kunnen begrijpen.
* Schema’s, tekeningen, PCB layouts, flowcharts enzovoort.
* Issues List: een (dynamische) lijst van alle openstaande onduidelijkheden t.a.v. het ontwerp, informatie waar nog op gewacht wordt en dergelijke.