|  |
| --- |
| Gravity Falls  TestPlan |

Versie informatie

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Datum** | **Bijzonderheden** | **Auteur** |
| 1.0 | 05-01-2016 | Start van het Document | Wout Kamp |
|  |  |  |  |

Verzendlijst

|  |  |
| --- | --- |
| **Naam** | **Bedrijf/Functie** |
| Patrick Beer | NetGames |
|  |  |

Inhoudsopgave

Versie informatie II

Verzendlijst III

Inhoudsopgave IV

1 Inleiding 1

1.1 Doel van het testplan 1

1.2 Opdrachtformulering 1

1.2.1 Opdrachtgever 1

1.2.2 Opdrachtnemer 1

1.2.3 Opdracht 1

1.2.4 Acceptanten en acceptatiecriteria 1

2 Testbasis 3

3 Productrisicoanalyse 4

4 Teststrategie / testaanpak 6

4.1 Teststrategie 6

4.2 Testproducten 6

5 Begrippenlijst 7

# Inleiding

## Doel van het testplan

Het doel van dit testplan (TP) is om voor het NetGames aan te geven wat de testaanpak is. In dit plan wordt aangegeven voor de volgende test soorten

<Unit test>  
Hier worden elke methodes in de applicatie getest. Hier wordt gekeken naar de kwaliteit van code.

<Component test>  
We gaan componenten in isolatie testen. Stub en drivers worden getest met gebruik van relatie tot ander componenten.

<Integratie test>  
Testen van het deelsysteem. Het testen van communicatie tussen componenten. Gebruik maken van stubs en/of drivers op interfaces van te testen deelsysteem. En eventueel testen van integratie van software met hardware.

<Systeem test>  
Testen van het volledige systeem.  
Is gebaseerd op systeem requirements met de hand van functionele en niet-functionele eisen.  
Wordt getest op uiteindelijke hardware.

<Acceptatie test>  
Test uitgevoerd door de opdradchtgever wordt getest in de omgeving van de klant.

In dit document wordt ieder die betrokken is bij de diverse testactiviteiten geïnformeerd over de aanpak, de activiteiten en de op te leveren producten met betrekking tot de verschillende testsoorten van project <identificatie project en/of opdracht>.

## Opdrachtformulering

### Opdrachtgever

NetGames

### Opdrachtnemer

<Fontys FHICT S31T Groep A>

De verantwoordelijken van deze testen zijn

Asror Wali  
Mehmet Bakirci  
Wout Kamp

### Opdracht

Het eindresultaat zal een functionele stabiele game zijn die op verschillende platformen te spelen zal zijn. Er moeten geen onverwachte problemen opduiken die het spelen van het spel kunnen belemmeren.

### Acceptanten en acceptatiecriteria

#### Acceptanten

Acceptanten namens de (opdracht)gevende organisatie zijn:

|  |  |  |
| --- | --- | --- |
| **Naam** | **Functie** | **Afdeling** |
|  |  |  |
|  |  |  |
|  |  |  |

#### Acceptatiecriteria

De acceptatiecriteria voor de systeemtest zijn:

|  |  |
| --- | --- |
| **Omschrijving** | **Norm** |
|  |  |
|  |  |
|  |  |
|  |  |

# Testbasis

In dit hoofdstuk wordt de gebruikte informatie (=testbasis) beschreven.

<< Hier wordt verwezen naar de informatie (hoeven niet altijd documenten te zijn!) die als uitgangspunt dient bij het testen. Denk hierbij aan Requirements, Use Cases, User Stories, Features, Functioneel of Technisch Ontwerp, datamodellen, systeemarchitectuur, handleidingen, ‘oude’ testware en AO-procedures.

*Voor het 3e semester is focus op systeem testen, unit testen en bij voorkeur ook component-testen. Geef daarom hieronder de documenten waarin de informatie over systeem requirements, component decompositie en implementatie klassen gevonden kan worden.* >>

De testbasis bestaat uit die informatie (vaak documenten) waaruit de testgevallen worden afgeleid. Onderstaand overzicht geeft de documentatie die als uitgangspunt dient bij de systeemtest.

|  |  |  |  |
| --- | --- | --- | --- |
| **Documentnaam** | **versie** | **Datum** | **Auteur** |
|  |  |  |  |
|  |  |  |  |

# Productrisicoanalyse

<< De beschikbare tijd om te testen is beperkt; niet alles kan even zwaar worden getest. Dus moesten er keuzes worden gemaakt. Daarbij is ernaar gestreefd om de testcapaciteit zo effectief en efficiënt mogelijk over het totale testtraject te verdelen.

De teststrategie legt vast *wat* er *met welke zwaarte* getest gaat worden en is er op gericht om zo vroeg mogelijk de belangrijkste fouten te vinden tegen de minste kosten, dus met optimaal gebruik van de beschikbare capaciteit en tijd. >>

De eerste stap bij het opstellen van de teststrategie is het uitvoeren van een *productrisicoanalyse*.

In overleg met de opdrachtgever en andere betrokkenen zijn de productrisico’s vastgesteld.

De mate van risico (de risicoklasse) is daarbij van de ene kant afhankelijk van de faalkans (hoe groot is de kans dat het fout gaat?) en van de andere kant van de schade voor de organisatie als het inderdaad fout gaat.

De risicoklasse (RK) is vervolgens bepalend voor de zwaarte van de test. Hierbij is risicoklasse A de hoogste risicoklasse en C de laagste. De teststrategie is er vervolgens bovendien op gericht om de risico’s met de hoogste risicoklasse zo vroeg mogelijk in het testtraject af te dekken.

Productrisicoanalyse (PRA) wordt bepaald door per testdoel te bepalen: faalkans, schade en risicoklasse.

<<Geef per testsoort een risicotabel. Testdoelen variëren per testsoort. Zie onderstaande tabellen als voorbeeld voor systemtesten en unittesten. Bij unitesten kan bijvoorbeeld module A een authenticatie module zijn, en module B een GUI module>>

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdoel** | **Schade** | **Faalkans** | **Risico klasse** |
| <<Actie 1>> | 3 | 3 | 9 |
| <<Actie 2>> | 3 | 2 | 6 |
| <<Rule 1>> | 3 | 3 | 9 |
| ….. | .. | .. | .. |
| <<Non Functionals 1 >>  Bijv. Gebruikers vriendelijkheid | 2 | 2 | 4 |

***Tabel: Productrisicotabel Systeem Testen <<voorbeeld>>***

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdoel** | **Schade** | **Faalkans** | **Risico klasse** |
| <<Module A>> | 3 | 2 | 6 |
| <<Module B>> | 1 | 1 | 1 |
| <<Module C>> | 3 | 1 | 3 |
| ….. | .. | .. | .. |
| << Library 1 >> | 2 | 2 | 4 |

***Tabel: Productrisicotabel Unit Testen. <<voorbeeld>>***

# Teststrategie / testaanpak

In dit hoofdstuk wordt, op basis van de productrisicoanalye, de teststrategie (het *wat*) opgesteld en vertaald naar een concrete testaanpak (het *hoe*).

## Teststrategie

<< Geef aan hoe je denkt de risico’s per Testdoel (zie hoofdstuk 3) te testen c.q. af te dekken. Beschrijf alleen hoe je dit denkt aan te pakken. Uitvoeren is geen onderdeel van de opdracht.>>

Voor elk risico uit de productrisicoanalyse is de risicoklasse bepalend voor de zwaarte van de test. Hierbij is risicoklasse A de hoogste risicoklasse en C de laagste. De teststrategie is er vervolgens bovendien op gericht om de risico’s met de hoogste risicoklasse zo vroeg mogelijk in het testtraject af te dekken.

Opzet: Bij zwaardere risicoklasse wil je meer testsoorten en per testsoort een zwaardere test inzetten. Geef dit in onderstaande tabel aan met een bolletje of sterretje. Bijvoorbeeld:

Risicoklasse A moet in minimaal één testsoort ●●● hebben, risicoklasse B in minimaal één testsoort ●● en risicoklasse C in minimaal één testsoort ● >

<< **Let op**: In deze tabel wordt een aantal testsoorten genoemd, maar slechts bij wijze van voorbeeld. In jouw project kan sprake zijn van meer/minder en/of andere en/of anders genoemde testsoorten. *Ga voor 3e semester uit van minimaal unit testen en automatische en handmatige systeemtesten. Bij voorkeur ook Component Testen, waarmee je de afzonderlijke testen in een gedistribueerd systeem test.*>>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Testdoel** | **PRA** | **UT** | **CT** | **IT** | **ST handm.** | **ST autom.** |
| Functionaliteit | A/B/C |  |  |  |  |  |
| - deel 1 |  |  |  |  |  |  |
| - deel 2 |  |  |  |  |  |  |
| - totaal |  |  |  |  |  |  |
| Gebruiksvriendelijkheid |  |  |  |  |  |  |
| Performance |  |  |  |  |  |  |
| - online |  |  |  |  |  |  |
| - batch |  |  |  |  |  |  |
| Beveiliging |  |  |  |  |  |  |
| Inpasbaarheid |  |  |  |  |  |  |

<<Toelichting bij de bovenstaande tabel:

|  |  |
| --- | --- |
| PRA-RK | Risicoklasse uit de productrisicoanalyse (PRA): risicotabel |
| Toetsen | Toetsing/review van de verschillende tussenproducten zoals requirements, functioneel ontwerp, technisch ontwerp |
| Ontwikkeltest | Unittest en Unitintegratietest |
| ST | Systeemtest |
| IT | Integratie test |
| CT | Component test |
| UT | Unit Test |
| ● | beperkte dynamische test |
| ●● | gemiddelde dynamische test |
| ●●● | zware dynamische test |

>>

## Testproducten

De op te leveren producten zijn in onderstaande tabel per fase te vinden.

|  |  |  |
| --- | --- | --- |
| **Fase** | **Product** | **Toelichting** |
| <Planning> | <Testplan> |  |
|  |  |  |
| <Realisatie> | <Testegevallen testsoort 1> |  |
|  | <Unittests> |  |
| <Test/Afronding> | <Test rapport testssoort 1> |  |
|  | <Test rapport tessoort 2> |  |
|  |  |  |
|  |  |  |

## Reviewplan

Geef een opsomming van de op te leveren producten door het testteam welke dienen te worden gereviewd.

<< Hierin dienen in elk geval onderstaande producten opgenomen te zijn:

* Systemtestplan
* Testgevallendocument
* Testrapport
* >>

| **Producten** | **Opstellers** | **Type review** | **Reviewers** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Go/No go

<< Beschrijf hier het Go/No go beslissingsproces, bijvoorbeeld:

Na afronding van het gehele testtraject wordt een vrijgaveadvies opgesteld. Hierin staan de resultaten van de verschillende testsoorten weergegeven en waarin, op basis daarvan, aangegeven wordt welke risico’s gelopen worden bij in productie name van het <naam systeem> systeem. Ook wordt hierin vermeld of er aan de gestelde acceptatiecriteria is voldaan.

Op basis van dit vrijgaveadvies kan door de acceptanten een Go/No go beslissing genomen worden.

Voor een Go zal voldaan moeten worden aan de eisen zoals gesteld in de paragraaf over de acceptatie criteria.>>

# Begrippenlijst

|  |  |
| --- | --- |
| PRA | Productrisicoanalyse, het analyseren van het te testen product met als doel dat de testmanager en de verschillende andere belanghebbenden tot een gezamenlijk beeld komen van wat de meer of minder risicovolle delen van het te testen product zijn, zodat de grondigheid van testen hieraan gerelateerd kan worden. |
| ST | Systeemtest, door de leverancier van de oplossing in een (goed beheersbare) laboratoriumomgeving uitgevoerde test, die moet aantonen dat de ontwikkelde systeem of delen daarvan aan de functionele- en niet-functionele specificaties en het technisch ontwerp voldoen.  << Zelf verder aan te vullen >> |