Test Rapport

Datum : 31 – 10 – 2018

Plaats : Arnhem

Vak : SAQ (System Analysis and Quality)

Naam Leraar : Frans Huijgen

Namen Auteurs : Joost Konst, Bram Soutendam en Brighton van den End

Student nummers : 597818, 599164, 608472

Versie : 5

# Inleiding Testrapport

In dit verslag hebben wij verschillende Use Cases uitgewerkt met behulp van activity diagrams, test cases en scenario’s en een onderzoek gedaan naar methoden voor softwareontwikkeling, hiervoor hebben we gebruik gemaakt van meerdere onderzoeksstrategieën zoals beschreven in het ICT Research Methods Pack kaartspel.

Inhoud

[Inleiding Testrapport 2](#_Toc528849119)

[Registreren verkoop 4](#_Toc528849120)

[Activity diagram 4](#_Toc528849121)

[Scenario’s 5](#_Toc528849122)

[Test cases 6](#_Toc528849123)

[Ontvangen goederen 8](#_Toc528849124)

[Activity diagram 8](#_Toc528849125)

[Scenario’s 9](#_Toc528849126)

[Test cases 10](#_Toc528849127)

[Opstellen besteladvies 12](#_Toc528849128)

[Activity diagram 12](#_Toc528849129)

[Scenario’s 13](#_Toc528849130)

[Test cases 14](#_Toc528849131)

[Inleiding Onderzoeksrapport 15](#_Toc528849132)

[Hoofd- en deelvragen 16](#_Toc528849133)

[Deelvraag 1 16](#_Toc528849134)

[Deelvraag 2 17](#_Toc528849135)

[Deelvraag 3 21](#_Toc528849136)

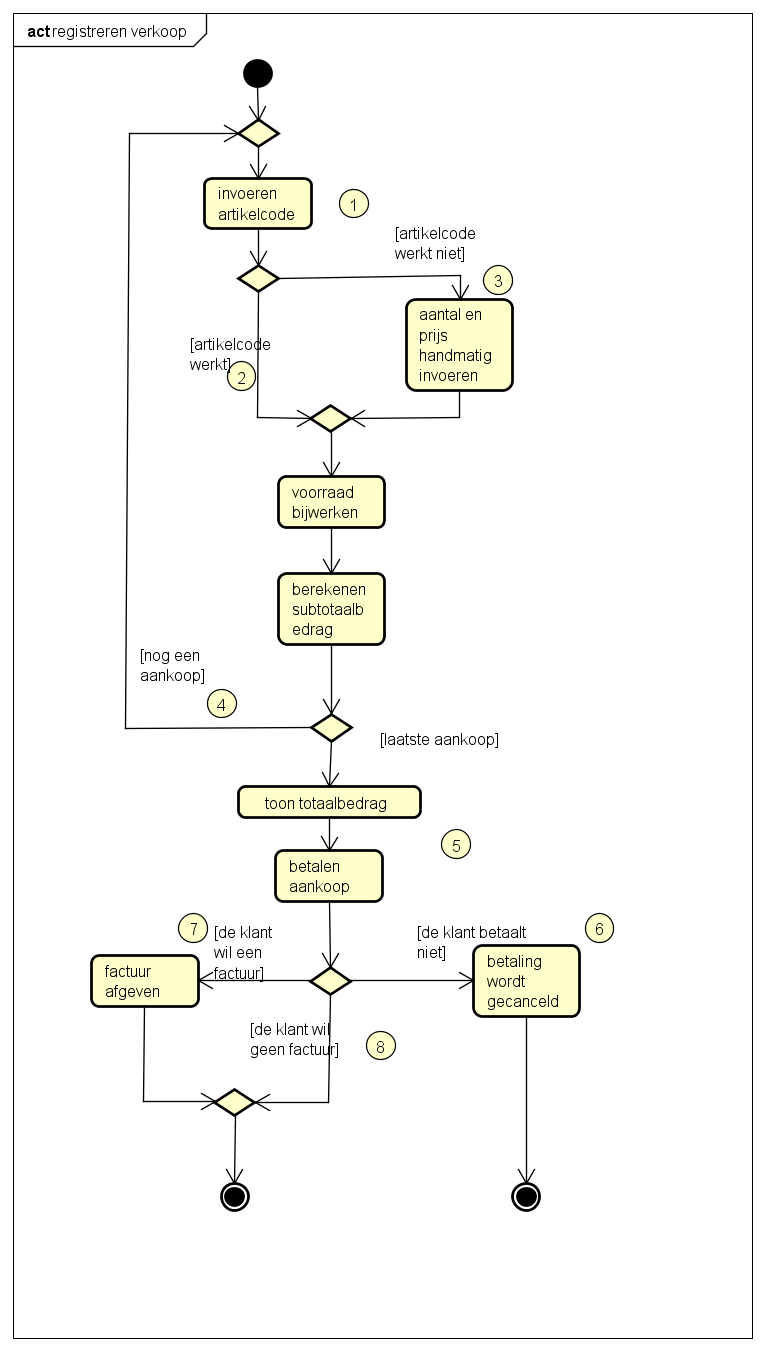
[Conclusie 22](#_Toc528849137)

[Bibliografie 23](#_Toc528849138)

[Bijlagen 24](#_Toc528849139)

# Registreren verkoop

## Activity diagram



## Scenario’s

**Onderscheiden paden:**

1-2, 1,3

2-4, 2-5

3-4, 3-5

4-1,

5-6, 5-7, 5-8

**Deelpaden:**

1. 1,2,4,3,5,7
2. 1,3,5,6
3. 1,2,5,8
4. 1,3,5,7
5. 1,2,5,6
6. 1,2,5,7
7. 1,2,5,8
8. 1,2,4,3,5,6
9. 1,2,4,3,5,8
10. 1,3,4,2,5,7
11. 1,3,4,2,5,8

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1,2 | 1,3 | 2,4 | 2,5 | 3,4 | 3,5 | 4,1 | 5,6 | 5,7 | 5,8 |
| 1 | x | x | x |  |  | x | x |  | x |  |
| 2 |  | x |  |  |  | x |  | x |  |  |
| 3 | x |  |  | x |  |  |  |  |  | x |
| 4 |  | x |  |  |  | x |  |  | x |  |
| 5 | x |  |  | x |  |  |  | x |  |  |
| 6 | x |  |  | x |  |  |  |  | x |  |
| 7 | x |  |  | x |  |  |  |  |  | x |
| 8 | x | x | x |  |  | x | x | x |  |  |
| 9 | x | x | x |  |  | x | x |  |  | x |
| 10 | x | x |  | x | x |  | x | x |  |  |
| 11 | x | x |  | x | x |  | x |  | x |  |
| 12 | x | x |  | x | x |  | x |  |  | x |

*Tabel 1 Matrix registreren verkoop*

scenario’s 1, 3 en 10 zorgen ervoor dat testmaat 1 wordt voldaan.

## Test cases

|  |  |  |
| --- | --- | --- |
| Legenda  Naam | Voorbeeld | Beschrijving |
| Artikel code | 00658842 | Een geldige artikel code bestaat uit acht nummers. |
| Rekening nummer | INGB NL3 1234 5678 | Een rekening nummer moet voldoen aan de IBAN nummer notatie. |
| Pin nummer | 1234 | Een heel nummer van vier tot en met zes cijfers. |
| Saldo | € 50,00 | Een nummer dat groter is dan 0 |
| Klant wil factuur | Ja | Een ja of nee. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario: iedere  werkende artikelcode, niet werkende artikelcode te weinig cijfers, niet werkende code letters | | Pad: ieder pad Activiteit 1 invoeren artikelcode | | |
| Stap | **Variabele** | **TC-1** | **TC-2** | **TC-3** |
| Invoer |  | | | |
|  | Artikel code | 00698842 | 0069998 | 000pcp9\_+ |
| Uitvoer | | | | |
|  | Artikel code | Hij toont de prijs van alleen dit product | Melding : ”je code is een cijfer te kort, heb je er een gemist?” | Melding : ”dit is een incorrecte code, vraag een medewerker u handmatig te helpen” |

*Tabel 4 Registreren verkoop 1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario: iedere  Verkeerd rekening nummer,  Verkeerd pinnummer, ontoereikend saldo. | | Pad: ieder pad Activiteit 2: betaling uitvoeren | | |
| Stap | **Variabele** | **TC-1** | **TC-2** | **TC-3** |
| Invoer |  | | | |
|  | Rekening nummer | Ingb nl3 0000 0000 00 | Ingb nl3 0000 0000 00 | 45687 |
|  | Pin nummer | 7777 | 7\*79 | 7777 |
|  | Saldo | € 0,00 | € 180,00 | € 180.00 |
| Uitvoer | | | | |
|  | Rekening nummer | Hij zou naar het volgende scherm moeten gaan | Hij zou naar het volgende scherm moeten gaan | Melding : ”dit is geen geldig rekeningnummer” |
|  | Pin nummer | Hij zou naar het volgende scherm moeten gaan | Melding : ”dit is geen correcte pincode” | hij zou naar het volgende scherm moeten gaan |
|  | Saldo | Melding : ”saldo ontoereikend” | Melding “dank u voor uw klandizie” | Melding : ”dank u voor uw klandizie” |

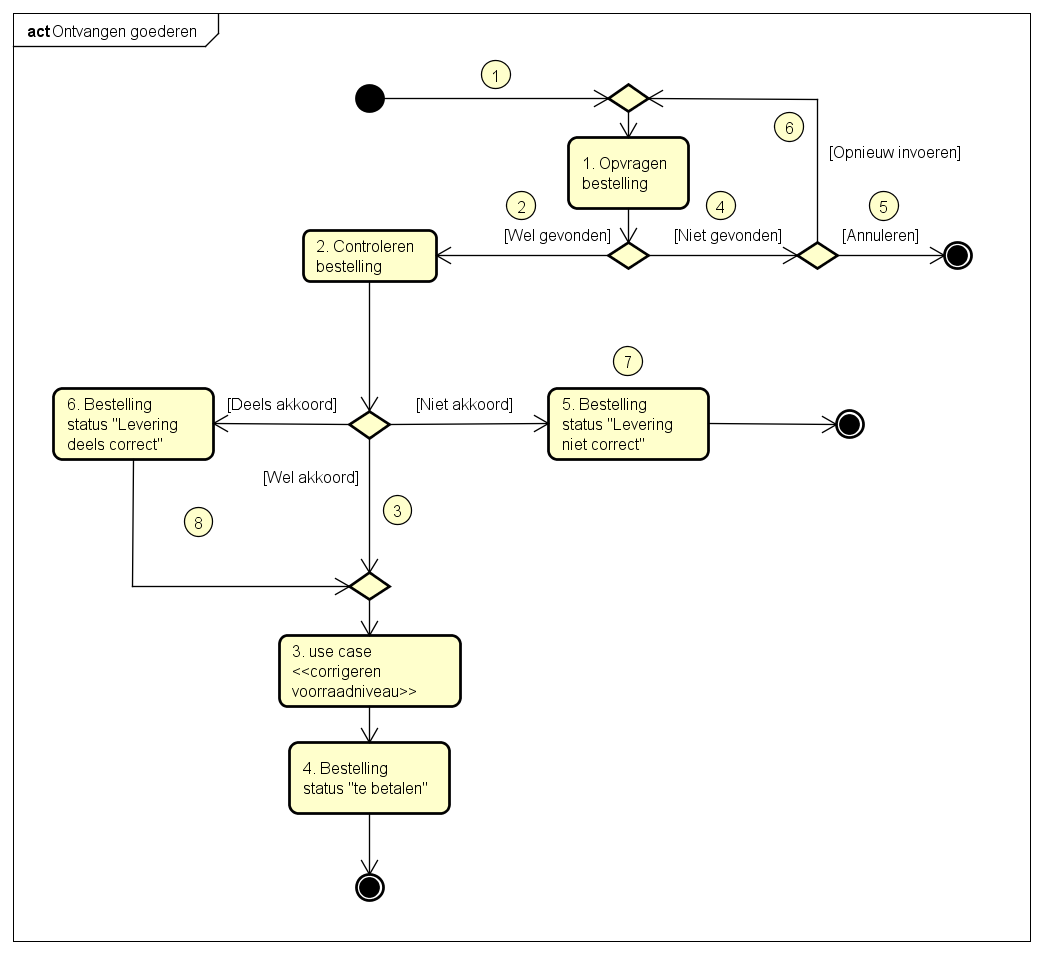
*Tabel 5 Registreren verkoop 2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario 1 en 3 | | Pad: Activiteit 3 factuur of niet | | |
| Stap | **Variabele** | **TC-1** | **TC-2** | **TC-3** |
| Invoer |  | | | |
|  | Klant wil factuur | ja | nee | Sluit de site zonder te antwoorden of hij een factuur wil |
| Uitvoer | | | | |
|  | Klant wil factuur | De factuur aan de klant gegeven | Er wordt geen factuur uitgeprint | Er wordt geen factuur uitgeprint |

*Tabel 6 Registreren verkoop 3*

# Ontvangen goederen

## Activity diagram



## Scenario’s

**Onderscheiden paden:**

1-2, 1-4

2-3, 2-7, 2-8

4-5, 4-6

6-2, 6-4

**Deelpaden:**

1. 1,2,3
2. 1,2,7
3. 1,2,8
4. 1,4,5
5. 1,4,6,2,3
6. 1,4,6,2,7
7. 1,4,6,2,8
8. 1,4,6,4,5

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1-2 | 1-4 | 2-3 | 2-7 | 2-8 | 4-5 | 4-6 | 6-2 | 6-4 |
| 1 | x |  | X |  |  |  |  |  |  |
| 2 | x |  |  | X |  |  |  |  |  |
| 3 | x |  |  |  | X |  |  |  |  |
| 4 |  | x |  |  |  | X |  |  |  |
| 5 |  | x | X |  |  |  | x | x |  |
| 6 |  | x |  | X |  |  | x | x |  |
| 7 |  | x |  |  | X |  | x | x |  |
| 8 |  | x |  |  |  | X | x |  | x |

*Tabel 2 Matrix ontvangen goederen*

Scenario’s 1, 2, 7 en 8 zorgen ervoor dat er aan testmaat 1 wordt voldaan.

## Test cases

|  |  |  |
| --- | --- | --- |
| Legenda  Naam | Voorbeeld | Omschrijving |
| Artikel code | 00658842 | Een geldige artikel code bestaat uit acht nummers. |
| Bestellijst | { 00557515 , 74583589 } | Een lijst met codes van alle te bestellen artikelen |

Variabele artikelnummer wordt gebruikt voor het opvragen van een bestelling. Een correct artikelnummer bestaat uit acht cijfers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario: 1  Opvragen bestelling, controleren bestelling, akkoord, bestelling status op “te betalen” | | Pad: 1,2,3   Activiteit 1: Opvragen bestelling | | |
| Stap | **Variabele** | **TC-1** | **TC-2** | **TC-3** |
| Invoer |  | | | |
|  | Artikel code | 58472908 | 58A7290B | 43258290 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Uitvoer | | | | |
|  |  |  |  |  |
|  | Artikel code | “Artikel met gegeven artikelnummer is gevonden.” | “Artikelnummer is onjuist.” | “Artikel met gegeven artikelnummer is niet gevonden, wilt u het opnieuw proberen of annuleren?” |
|  |  |  |  |  |
|  |  |  |  |  |

*Tabel 7 Ontvangen goederen 1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario: 4  Opvragen bestelling, niet gevonden, annuleren | | Pad: 1,4,5   Activiteit 1: Opvragen bestelling | | |
| Stap | **Variabele** | **TC-1** | **TC-2** | **TC-3** |
| Invoer |  | | | |
|  | Artikel code | 58472908 | 58A7290B | 43258290 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Uitvoer | | | | |
|  |  |  |  |  |
|  | Artikel code | “Artikel met gegeven artikelnummer is gevonden.” | “Artikelnummer is onjuist.” | “Artikel met gegeven artikelnummer is niet gevonden, wilt u het opnieuw proberen of annuleren?” |
|  |  |  |  |  |
|  |  |  |  |  |

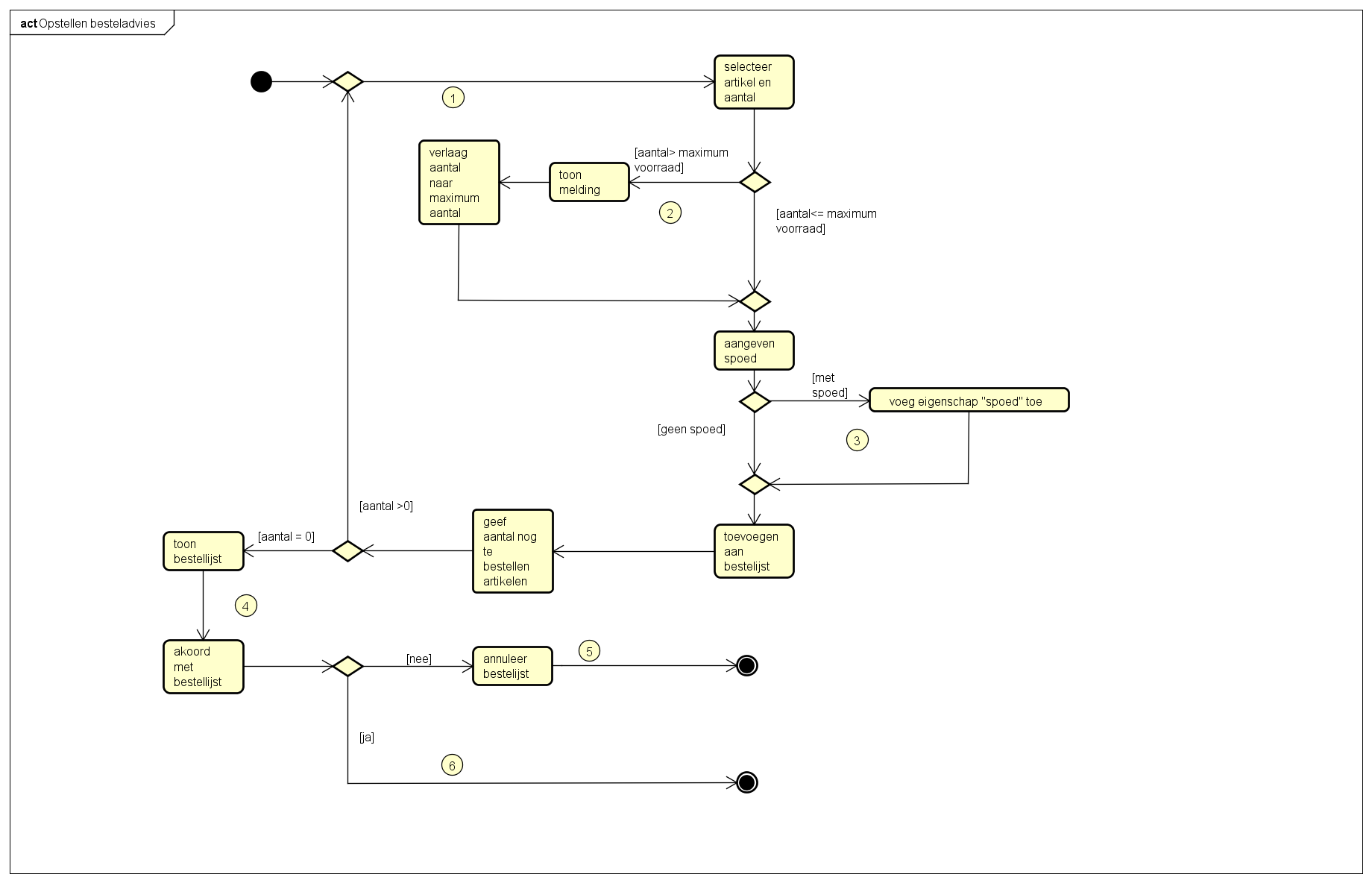
*Tabel 8 Ontvangen goederen 2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Scenario: 3  Opvragen bestelling, controleren bestelling, deels akkoord, bestelling status op “levering deels correct” | | Pad: 1,2,8   Activiteit 2: Controleren bestelling | | |
| Stap | **Variabele** | **TC-1** | **TC-2** | **TC-3** |
| Invoer |  | | | |
|  | Bestellijst | { 00557515 , 74583589 } | { 00557515 , 7489 } | { 0515 , 74fe2898} |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Uitvoer | | | | |
|  | Bestellijst | Bestelling status op “te betalen” | Bestelling status op “Levering deels correct” | Bestelling status op “Levering niet correct” |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

*Tabel 9 Ontvangen goederen 3*

# Opstellen besteladvies

## Activity diagram



## Scenario’s

**Onderscheiden paden:**

1 – 1 , 1 – 2 , 1 – 3 , 1 – 4

2 – 3 , 2 – 4 , 2 – 1

3 – 1 , 3 – 4

4 – 5 , 4 – 6

**Deelpaden:**

1. 1 -> 1 -> 2 -> 3 -> 4 -> 5
2. 1 -> 1 -> 2 -> 3 -> 4 -> 6
3. 1 -> 4 -> 5
4. 1 -> 4 -> 6

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario | 1-1 | 1-2 | 1-3 | 1-4 | 2-3 | 2-4 | 2-1 | 3-1 | 3-4 | 4-5 | 4-6 |
| 1 | X | X |  |  | X |  |  |  | X | X |  |
| 2 | X | X |  |  | X |  |  |  | X |  | X |
| 3 |  |  |  | X |  |  |  |  |  | X |  |
| 4 |  |  |  | X |  |  |  |  |  |  | X |

*Tabel 3 Matrix opstellen bestellijst*

Scenario’s 1 en 2 zorgen ervoor dat er aan testmaat 1 wordt voldaan.

## Test cases

|  |  |  |
| --- | --- | --- |
| Legenda  Naam | Voorbeeld | Omschrijving |
| Artikel code | 00658842 | Een geldige artikel code bestaat uit acht nummers. |
| Bestellijst | { 00557515 , 74583589 } | Een lijst met codes van alle te bestellen artikelen |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Scenario: 1  Opstellen besteladvies | | Pad :  1 -> 1 -> 2 -> 3 -> 4 -> 6 | | Activatie :  Toevoegen aan bestellijst | |
| Stap | **Variabele** | **Test Case 1** | **Test Case 2** | | **Test Case 3** |
| Invoer |  | | | | |
|  | Artikel code | 00655589 | 00655589 | | 00f\_55\*+ |
|  | Is spoed | Ja | Nee | | Nee |
|  | Bestellijst | - | - | | - |
| Uitvoer | | | | | |
|  | Artikel code | 93841 | 92841 | | Melding : Artikel code is geen valide code |
|  | Is spoed | Ja | Nee | | Nee |
|  | Bestellijst | Artikel is toegevoegd | Artikel is NIET toegevoegd | | Artikel is NIET toegevoegd |

*Tabel 10 Opstellen besteladvies 1*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Scenario: 1  Opstellen besteladvies | | Pad :  1 -> 1 -> 2 -> 3 -> 4 -> 6 | | Activatie :  Akkoord bestellijst | |
| Stap | **Variabele** | **Test Case 1** | **Test Case 2** | | **Test Case 3** |
| Invoer |  | | | | |
|  | Akkoord | Ja | Nee | | Ja |
|  | Bestellijst | Aanwezig | Aanwezig | | Niet aanwezig |
| Uitvoer | | | | | |
|  | Akkoord | Ja | Nee | | - |
|  | Bestellijst | Melding : Bestellijst wordt besteld | Melding : Bestellijst wordt NIET besteld | | Melding : Geen bestellijst aanwezig |

*Tabel 11 Opstellen besteladvies 2*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Scenario: 2  Opstellen besteladvies | | Pad :  1 -> 1 -> 2 -> 3 -> 4 -> 6 | | Activatie :  Aantal artikelen te bestellen | |
| Stap | **Variabele** | **Test Case 1** | **Test Case 2** | | **Test Case 3** |
| Invoer |  | | | | |
|  | Bestellijst | Aantal is meer dan | Aantal is gelijk aan 0 | | Aantal is minder dan 0 |
| Uitvoer | | | | | |
|  | Bestellijst | Melding : Meer artikelen te bestellen | Melding : geen artikelen nog te bestellen | | Melding : Aantal is negatief. |

*Tabel 12 Opstellen besteladvies 3*

Inleiding Onderzoeksrapport

Het bedrijf IT Products is over de laatste jaren flink gegroeid, waar hun ooit twee mensen in een klein kantoor waren, bestaan hun nu uit 18 mensen. Maar helaas is hun productiviteit aanzienlijk minder gegroeid dan hun aantal personeel, en ze merken op dat wanneer ze hun projecten af hebben de klant niet tevreden is. Het overduidelijke probleem is dat IT products hun software development methoden niet hebben mee ontwikkeld. Daarom hebben wij in dit hoofdstuk drie van de meest populaire software development methoden uiteengezet.

Hoofd- en deelvragen

De hoofdvraag luidt: Welke methode voor softwareontwikkeling kan het best gebruikt worden voor GameParadise?

De deelvragen luiden:

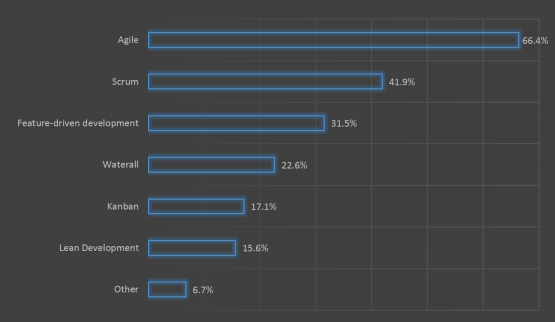
1. Wat zijn de drie meest voorkomende methoden voor software ontwikkeling in de markt?
2. Wat zijn de kenmerken van deze methoden?
3. Welke methoden kan het best ingezet worden bij welk type project?

Deelvraag 1

Voor het beantwoorden van deelvraag 1: Wat zijn de drie meest voorkomende methoden voor software ontwikkeling in de markt?, zal gebruik gemaakt worden van de onderzoeksstrategie veld. Binnen deze strategie is gekozen voor de methode Survey. Deze methode is gekozen omdat dit de enige manier is om te zien welke software ontwikkelingsmethode de meest gebruikte is.

De drie meest voorkomende softwareontwikkeling methoden zijn als volgt:

1. Waterval
2. Scrum
3. Lean



*Figuur 1: Meest voorkomende softwareontwikkeling methoden (Assembla, Usersnap, 2016)*

Deelvraag 2

Voor het beantwoorden van deelvraag 2: Wat zijn de kenmerken van deze methoden?, zal gebruik gemaakt worden van de onderzoeksstrategie bieb. Binnen deze strategie is gekozen voor de methode Literature study. Deze methode is gekozen omdat de kenmerken zelf feiten zijn, en deze zijn al gedocumenteerd dus op te zoeken.

**Waterval**

De waterval methode was vroeger veruit de meest populaire software development methode maar neemt al een poos een lagere plek in op de populariteitsladder. De reden hiervoor is dat de watervalmethode gemaakt is om de ene stap na de ander te uit te voeren. Dit is een van de grote voordelen van de watervalmethode maar tegelijkertijd ook een belangrijk nadeel. (S. Baliji, 2012)

De watervalmethode wordt voornamelijk gekenmerkt door deze attributen:

* Het waterval model is een sequentieel ontwikkelingsmodel
* Iedere volgende fase begint pas als de vorige eindigt
* De verschillende fases van een watervalmodel overlappen niet
* De taken binnen iedere fase zijn ingepland om op een specifieke tijd klaar te zijn
* The documentatie vindt plaats aan het eind van iedere fase
* Iedere fase stopt compleet vóór de volgende fase begint
* De rol van de testers begint pas in de testfase

Een groot voordeel aan de waterval methode is dat alle requirements al voordat de designfase begint duidelijk zijn, dit zorgt ervoor dat iedereen die er aan werkt al vrijwel meteen weet wat het uiteindelijke doel is. Een ander belangrijk voordeel aan de waterval methode is dat iedere fase op een specifiek moment eindigt, dit geeft een concreter beeld van waar het project is aan de klant. Verder is het door zijn lineaire model en er maar een specifiek team van bijv. designers aan werken, maakt het een erg makkelijk model om te implementeren.

Er zitten natuurlijk ook nadelen aan, het is bijvoorbeeld zo dat als de cliënt een extra requirement wil toevoegen na de requirementsfase dit vaak niet gebeurt. Dit is zo omdat wanneer de requirementsfase eindigt er gewoon geen nieuwe requirements bijkomen, voor dit model behoort alles in een specifieke fase en een gesloten fase open je niet opnieuw. Verder worden, omdat het testen niet tijdens het hele project plaatsvindt, er vaak problemen niet gevonden voor het einde van die fase. Dit soort problemen kunnen nog worden opgelost maar dat is dan al veel moeilijker geworden.

**Scrum**

**Inleiding**

Scrum is een agile werkmanier om een project succesvol, efficiënt en flexibel te realiseren in teamverband. De scrum werkmethode is al zeer succesvol in de IT wereld en inmiddels wordt deze werkwijze ook in de business afdelingen zoals Marketing, Sales, Human Resources en Finance gebruikt. In de kern van scrum staat een zelfsturend en multidisciplinair team centraal. Iedereen is betrokken bij het plannen en het verdelen van taken. Bij scrum wordt ervan uitgegaan dat de kennis voor de taken en doelen in het projectgroep aanwezig zijn.

**Werking**

De scrum methode werkt User Stories te maken om daarmee doelen op te stellen die in sprits worden uitgevoerd. Het stellen van doelen worden niet alleen per sprint gesteld maar ook dagelijks. Een User Story beschrijft één functionaliteit van het project. Een sprint duurt minimaal twee tot maximaal 4 weken waarin er aan User Stories gewerkt wordt. Waarbij per sprint werkende functionaliteit wordt geleverd aan de klant dat beschreven staat in een User Story. Er wordt in korte iteraties ofwel sprints gewerkt om zo de gestelde doelen voor en sprint te halen. Doordat er in korte iteraties gewerkt wordt heeft de klant eerder en beter inzicht in de gang van zaken en is er een mogelijkheid om feedback te geven over de geleverde functionaliteiten.

**Voor- en nadelen**

Maar scrum is niet zonder nadelen.

|  |  |
| --- | --- |
| Voordelen | Nadelen |
| Iteratief | Niet eenvoudig |
| Snelle realisatie | Moeilijk te realiseren zonder grondig onderzoek |
| Snelle feedback verwerking | Veel tijd nodig van het bedrijf |
| Ieder teamlid weet wat er aan het project gebeurd | Lastig om ieder teamlid goed te informeren. |
|  | Individuele rollen onduidelijk |

**Lean-softwareontwikkeling**

De naam komt oorspronkelijk van een boek dat met dezelfde naam is geschreven in 2003 door Mary Poppendieck en Tom Poppendieck. In het boek zijn de traditionele lean principes beschreven samen met de 22 gereedschappen die volgens Mary en Tom Poppendieck hierbij gebruikt worden.

Lean software ontwikkeling bestaat uit zeven principes:

1. Voorkom verspilling
2. Versterk het leereffect
3. Beslis zo laat mogelijk
4. Lever zo snel mogelijk
5. Geef het team verantwoordelijkheid
6. Kwaliteit
7. Zie het geheel

**Voorkom verspilling**

Alles wat voor het uiteindelijke product geen waarde heeft wordt gezien als verspilling. Daar vallen de onderstaande dingen onder:

* Onduidelijke requirements
* Bureaucratie
* Langzame interne communicatie
* Vertragingen in het softwareontwikkelproces
* Onnodige code en functionaliteit
* Te weinig tests

Verspillingen moeten eerst herkend worden voordat ze verwijderd kunnen worden. Een bepaalde techniek wordt toegepast om zulke verspillingen te vinden en te verwijderen. Dit proces moet vaak een aantal keren herhaald moet worden voordat alle verspillingen ontdekt en verwijderd zijn.

**Versterk het leereffect**

Softwareontwikkeling is een continu leerproces wat alleen maar toeneemt met de samenwerking in teams. De beste manier om de softwareontwikkeling te verbeteren is door het leereffect te versterken. Het is beter om verschillende ideeën uit te proberen, zodat het sneller duidelijk wordt wat de wensen zijn van de gebruiker.

**Beslis zo laat mogelijk**

Het idee van dit principe is om eerst meerdere mogelijke oplossingsrichtingen te bedenken, zodat de uiteindelijke beslissingen niet op aannames maar op feiten zijn gebaseerd, zo is het mogelijk om onomkeerbare beslissingen zo laat mogelijk te maken. Daardoor is er minder een element van onzekerheid aanwezig en kan er geld bespaard worden door dure beslissingen later te maken.

**Lever zo snel mogelijk**

Des te sneller het product wordt opgeleverd des te sneller de klant feedback erover kan geven waar het team van kan leren. Hoe vaker het product opnieuw wordt opgeleverd, hoe sneller er gecommuniceerd kan worden tussen de klant en het team.

**Geef het team verantwoordelijkheid**

Binnen een bedrijf is het meestal zo dat de managers het personeel aansturen. Dat werkt alleen heel goed als de managers ook precies weten wat er gebeuren moet. Binnen de lean aanpak bevindt zich het gezegde: “vind goede mensen en laat hen het werk doen”, deze mensen moeten ook gemotiveerd worden, maar precies voorschrijven wat er gedaan moet worden werkt niet op deze manier.

**Kwaliteit**

Om een goede kwaliteit te behouden moeten de afzonderlijke onderdelen van een systeem goed met elkaar samenwerken. De informatie die van klant naar ontwikkelaar en terug moet continu gebeuren zodat de klant niet in een keer iets opgeleverd krijgt wat niet volgens de wensen is. Een manier om tot een product van goede kwaliteit te komen is refactoren. Dit houdt in dat alle extra dingen weg worden gehaald of versimpeld, bijvoorbeeld herhalingen voorkomen.

**Zie het geheel**

De meeste software systemen bestaan tegenwoordig uit meer dan één onderdeel, het is dus belangrijk hoe de onderdelen samenwerken. Dus door de grote onderdelen op de delen in kleine stukken wordt het overzichtelijker. “*Think big; act small; fail fast; learn rapidly*” (Mary Poppendieck, 2003)

**Voordelen**

Voordelen van de lean-software wikkelings methode zijn als volgt. Het team dat aan het product werkt leert veel tijdens het hele proces. De belangrijke beslissingen worden zo laat mogelijke gemaakt om zo onzekerheid te vermijden. Het product waar aan gewerkt wordt word zo snel mogelijk en zo vaak mogelijk opgeleverd aan de klant zodat de eisen van de klant duidelijker worden. Het team krijgt zelf veel verantwoordelijkheid als het gaat om wat er precies gebeurd moet worden. Om de kwaliteit te waarborgen wordt er continu informatie uitgewisseld tussen klant en ontwikkelaar en andersom, zodat de klant precies krijgt wat er gevraagd is.

**Nadelen**

Een nadeel van deze software ontwikkelings methode is dat als er zogeheten verspillingen optreden, wordt het hele proces langzamer. De requirements moeten duidelijk zijn, de communicatie intern moet snel verlopen, er moet voldoende getest worden, enzovoort.

Deelvraag 3

Voor het beantwoorden van deelvraag 3: Welke methode kan het best ingezet worden bij welk type project? Zal gebruik gemaakt worden van de onderzoeksstrategie bieb. Binnen deze strategie is gekozen voor de methode Literature study. Deze methode is gekozen omdat deze vraag al eerder beantwoord en onderzocht is door anderen studies is er besloten om de informatie van andere studies te gebruiken voor het beantwoorden van deze deelvraag.

**Waterval**

Als de requirements duidelijk zijn en het een erg groot project is het watervalmodel vaak een goede keuze, maar als de cliënt zelf nog niet zeker lijkt over zijn requirements is het toch vrij vaak een risico.

**Lean**

De lean software ontwikkelings methode kan het best ingezet worden bij een project waar van tevoren de requirements duidelijk zijn opgesteld, waar een of meerdere teams tegelijk aan het project kunnen werken.

**Scrum**

De scrum software ontwikkelings methode werkt het beste bij projecten waar de requirements nog onduidelijk zijn. En is ontworpen om snel op bijkomende of veranderende requirements te kunnen reageren en het project te kunnen aanpassen.

Conclusie

Omdat een van de voornaamste klachten over de huidige software ontwikkelings methoden was dat de ontwikkelingstijd te lang was, en het resultaat niet altijd aansloot bij de wensen van de klant, hebben wij besloten dat de meest passende software ontwikkelings methode voor bedrijf GameParadise de scrum methode zal zijn.

Via deze methode hoeft de klant niet meteen duidelijke requirements te stellen, want via deze iterative ontwikkelingsmethode kan er gaandeweg aangepast worden op de wensen van de klant. Ook heeft deze methode sneller een zichtbaar resultaat, dit helpt met het tonen van voortgang aan de klant.

De andere methoden zijn minder geschikt omdat de eerder genoemde klacht, dat het product niet altijd aansloot op de wensen van de klant, waarschijnlijk licht aan dat de requirements onduidelijk waren. Zouden de lean en waterval methodes minder geschikt zijn geweest want deze eisen duidelijke requirements om goed uitgevoerd te worden.

Bibliografie

Assembla, Usersnap. (2016). *The state of client collaboration in development & design.*

Mary Poppendieck, T. P. (2003). *Lean Software Development: An Agile Toolkit.* Upper Saddle River: Addison Wesley.

S. Baliji, M. S. (2012). Waterfall Vs V-model Vs agile: a comparative study on sdlc. *Internation Journal of Information Technology and Bussiness Management*, 26-30.

Swart, N. d. (2018, 11 1). *Agile kent ook deze 4 nadelen*. Opgehaald van Reaco Academy: https://www.reaco.nl/blog/agile-kent-ook-deze-4-nadelen/

*Wat is Scrum*. (2018, 11 1). Opgehaald van Prowareness: <https://www.scrum.nl/wat-is-scrum-agile-scrum/>

# Bijlagen

1. **SAQ Testrapport - feedbackformulier**

**Feedbackformulier voor : Lucas Folkers , Buncan Stevense , Jiankai Zheng**

**Feedback gegeven door : Joost Konst , Bram Soutendam , Brighton van den End**

Let op: bewaar dit feedbackformulier goed, dit is een verplichte bijlage bij je definitieve test rapport!

|  |  |  |
| --- | --- | --- |
|  | Aanwezig? (ja/nee) | Indien aanwezig “ja”:  Is het correct uitgevoerd? (indien niet correct: licht toe hoe het verbeterd kan worden) |
| Voorblad: |  |  |
| * Titel | Ja | - |
| * Naam/ Namen | Ja | - |
| * Studentnummer(s) | Ja | - |
| * Coursenaam | Ja | - |
| * Docentnaam | Ja | - |
| * Plaats & Datum | Ja | - |
|  |  |  |
| Optioneel: Titelblad en Voorwoord |  |  |
|  |  |  |
| Inhoudsopgave |  |  |
| * Hoofdstukken zijn genummerd | Ja | - |
| * Niveaus in hoofdstukken worden weergegeven (indien van toepassing) | Ja | De niveaus zijn te diep. De afspraak was max twee lagen diep |
| * Paginanummers zijn weergegeven | Ja | - |
| * Opbouw is overzichtelijk | Ja | - |
|  |  |  |
| Inleiding |  |  |
| * Beschrijft aanleiding voor het maken van het rapport | Nee | - |
| * Beschrijft de doelstelling van het rapport/de werkzaamheden | Ja | - |
| * Beschrijft de gekozen werkwijze | Ja | - |
| * Geeft aan welke randvoorwaarden gelden bij het maken van het rapport of het uitvoeren van het werk | Nee | - |
| * Geeft aan welke uitgangspunten van toepassing zijn | Ja | - |
| * Geeft de structuurbeschrijving van het rapport weer | Ja | - |
|  |  |  |
| Kern |  |  |
| * Ieder hoofdstuk start met inleiding | Ja | De inleiding is één zin. Dit is te kort. |
| * Heldere structuur, één onderwerp per alinea | Ja | Sommige tabellen zijn verdeelt over twee pagina’s |
| * Puntsgewijze opsommingen | Ja | - |
| * Indeling conform inhoudsopgave | Ja | - |
| * Correcte verwijzingen naar externe bronnen (auteur, jaartal) | Ja | - |
|  |  |  |
| Conclusie | Nee | Niet aanwezig |
| * Legt relatie naar volgend hoofdstuk of volgende activiteit | Nee |  |
| * Is geschreven in de tegenwoordige tijd | Nee |  |
| * Is gebaseerd op inhoud van het hoofdstuk of het rapport | Nee |  |
| * Is kernachtig verwoord | Nee |  |
| * Is startpunt voor aanbeveling | Nee |  |
|  |  |  |
| Aanbeveling | Nee | Niet aanwezig |
| * Sluit aan bij conclusie | Nee |  |
| * Is concreet, beschrijft wat er nu moet gebeuren | Nee |  |
| * Is uitvoerbaar | Nee |  |
|  |  |  |
| Verwijzingen | Nee | Niet aanwezig |
| * Pagina’s zijn genummerd | Nee |  |
| * Hoofdstukken hebben titels | Nee |  |
| * Figuren en tabellen hebben naam en nummer | Nee |  |
| * Verwijzingen naar bijlagen met nummer en titel van bijlage | Nee |  |
| * Kern van de bijlage wordt in de tekst weergegeven | Nee |  |
|  |  |  |
| Literatuurlijst volgens APA systematiek |  |  |
| * Boek: Auteur, A. (jaar van uitgave). Titel van het boek. Plaats: Uitgeverij. | Ja | - |
| * Internetbron: Auteur, A. (jaar van uitgave). Titel van het document. Geraadpleegd op dag maand jaar, van <http://url> | Nee |  |
|  |  |  |
| Bijlagen | Nee | Niet aanwezig |
| * Nummer of letter in titel | Nee |  |
| * Zelfstandig leesbaar | Nee |  |
|  |  |  |
| Verzorgd |  |  |
| * Leesbaar: zinnen ‘lopen’ | Ja | - |
| * Schrijfstijl gericht op doelgroep | Ja | - |
| * Taalkundig correct (in totaal < 3 spel- en grammaticale fouten op 1 pagina) | Ja | - |
| * Actief taalgebruik (=geen onnodige lijdende zinnen) | Ja | - |
| * Geen vage taal | Ja | - |

|  |  |  |
| --- | --- | --- |
| Inhoud correct? |  | Noteer hier *zelf* de kern van de inhoudelijke feedback je van je docent hebt gekregen |
|  |  |  |
| Tijdig opgeleverd? |  | Let op: dit is een knock out criterium voor zowel je use case- als je testrapport: niet op tijd ingeleverd wordt niet nagekeken |