

Het opstellen van use-cases

Inleiding

Dit document beschrijft algemene richtlijnen voor het opstellen van use-cases. Een use-case beschrijft de interactie tussen gebruikers en het te ontwikkelen systeem.

Voor het vastleggen van de software requirements worden de volgende twee soorten documenten gebruikt:

- Use-case model overzicht document: geeft een overzicht van alle betrokken use-cases en actoren
- Use-case specificatie document: per use-case, 1 use-case beschrijving.

Een use-case beschrijving bestaat dus uit meerdere documenten!

Definities, acroniemen en afkortingen

Begrip	Omschrijving
Actor	Een extern persoon, organisatie of systeem dat betrokken is in een use-case.
Primaire Actor	De actor die de use-case initieert.
Secundaire Actor	Een bij een use-case betrokken actor anders dan de primaire actor.
Requirements	De verzameling functionele en non-functionele eisen aan het systeem.
Use-case	Een manier om het systeem te gebruiken. Een beschrijving van de interactie (stappen) van één of meerdere Actoren met het systeem.
Use-case diagram	Grafische weergave van de interactie tussen use-cases en de betrokken actoren.
Scenario	Een use-case bestaat uit één of meerdere scenario's. Een use-case heeft tenminste één goed scenario waarmee de gewenste eindconditie wordt bereikt. Een scenario is feitelijk een bepaalde reeks (functionele) stappen.
Stakeholder	Stakeholders zijn gebruikers wiens belangen moeten worden behartigd bij het al dan niet succesvol uitvoeren van een use-case. Stakeholders zijn in de regel niet actief betrokken in een use-case. De belangen van stakeholders komen met name naar voren bij zaken als validaties en controles. Door het vastleggen van stakeholder doelstellingen worden impliciet belangrijke <i>business rules</i> gedocumenteerd.
Pre-conditie	Een beschrijving van de randvoorwaarden waaraan moet zijn voldaan om de use-case uit te mogen voeren.
Trigger	De actie (van buiten af) die de use-case start c.q. die de use-case veroorzaakt.
Frequentie	Het aantal keren dat een use-case wordt uitgevoerd per tijdseenheid.

Begrip	Omschrijving
Basic Flow of Events	Een beschrijving van het scenario dat wordt uitgevoerd wanneer zich geen foutsituaties of uitzonderingen voordoen. De eindsituatie wordt altijd expliciet genoemd (dus de status waarin de use-case zich bevindt na afloop). De eindconditie geeft aan welke belangen van welke stakeholders zijn gediend na het beëindigen van de use-case.
Actorkanaal	Een actorkanaal is een manier waarop een actor met het systeem kan communiceren, bijvoorbeeld GUI, berichten, post, telefoon, sockets, etc.
Include	Een include kan worden beschouwd als een (verplichte) sub-flow die bij meerdere use-cases voorkomt. Om schrijfwerk te voorkomen wordt de sub-flow als aparte (sub-) use-case gespecificeerd, die wordt aangeroepen vanuit de base use-case.
Extend	Een extend (<<extend>> relationship) is een alternatief scenario dat wordt uitgevoerd als een bepaalde conditie geldig is (if statement). Een extend kan een alternatieve stap zijn maar ook een complete (sub) use-case.
Alternative Flow	Een alternative flow kan worden gezien als een scenario dat wordt uitgevoerd bij fout- of uitzondering situaties. Ook hier wordt de eindsituatie (de status van de use-case) expliciet genoemd. Indien de flow bij meerdere use-cases een rol speelt, kan deze omgezet worden naar een <<extend>> use-case. Dit voorkomt redundantie.

Use-case model overzicht document

Hier wordt een opsomming gegeven van alle betrokken use-cases en actoren. Bevat een use-case overzichtsdigram dat alle use-cases bevat, inclusief de onderlinge relaties en de actoren. Daarnaast wordt een korte beschrijving gegeven per use-case en per actor.

Use-case specificatie

In een "Use-case specification" document worden de functionele specificaties van één use-case beschreven, 1 document per use-case. Per use-case dient het onderstaande formaat te worden toegepast.

Een use-case moet zijn voorzien van:

- een naam en een nummer, als unieke identificatie
- een korte beschrijving
- de primaire actor
- de trigger (waardoor wordt de use-case geïnitieerd)
- de basic flow of events, waarin sub-flows kunnen zijn opgenomen
- Alternative flows
- Precondities
- Gerelateerde gegevens (voor zover bekend)
- Aanvullende opmerkingen
- Openstaande punten

De naam van een use-case

- dient een indicatie te geven van welk doel wordt bereikt voor de actor
- dient uniek te zijn binnen het project
- wordt meestal samengesteld uit een werkwoord gevolgd door één of meerdere zelfstandig naamwoorden (bijv. "Toevoegen klantgegevens")

Actor

- Een actor moet zijn voorzien van een naam en een korte beschrijving
- De naam van een actor dient een zelfstandig naamwoord te zijn (bijv. 'Bankmedewerker') of de naam van een systeem (bijv. KISS).

Use-case diagram

- Een overzichtsdiagram dient alle use-cases van het systeem te tonen.
- Daarnaast dient elk use-case 1 diagram te bevatten met de daarvoor relevante actoren

Use-case specificatie

- Een use-case specificatie wordt primair vastgelegd middels gestructureerde tekst;
- Een use-case specificatie moet eenvoudig te lezen zijn;
- Een use-case specificatie moet een in staat stellen een gewenst (gebruikers) doel te realiseren;
- Een use-case heeft minimaal één actor (primaire actor);
- Een use-case specificatie bevat geen GUI elementen;
- Iedere use-case specificatie heeft:
 - pre-condities
 - één succes scenario (basis flow of events)
 - alternatieve scenario's
 - een eindstatus per scenario;
- De stappen moeten worden opgesteld in de tegenwoordige tijd. Bij iedere stap moet een Actor óf het systeem iets doen (niet beiden).
- De stappen van de basisstroom en de alternatieve stromen moeten worden genummerd.
- Elke alternatieve stroom wordt voorzien van een naam en een uniek volgnummer.
- In de alternatieve stroom wordt expliciet verwezen naar de stap uit de basisstroom waar het alternatief scenario kan optreden.
- *Minimale eindconditie:*
Als een use-case niet succesvol wordt beëindigd heeft dit doorgaans impact op de belangen van stakeholders; Bijvoorbeeld systeembeheerders, maar eventueel ook de leverancier van het systeem, of in het uiterste geval een aan de overheid gerelateerd orgaan. Minimale eindcondities zijn van belang bij het borgen van onder meer integriteit van gegevens, security aspecten, business rules, etc. De eindconditie wordt per scenario expliciet genoemd.

Een voorbeeld:

UC#:	UC01
Naam:	Beheer Studentengegevens
Omschrijving:	Deze use-case dient voor het toevoegen, wijzigen en verwijderen van studentengegevens, inclusief het beheer van behaalde cijfers per student.
Primaire Actor:	Studentenadministratie, hierna de gebruiker genoemd
Trigger:	Een medewerker van de studentenadministratie begint de verwerking van de studentenmutaties
Basic Flow:	
Stap	Actie
1	De gebruiker selecteert muteren studentengegevens
2	Het systeem toont een lijst met beschikbare studenten
3	De gebruiker selecteert eventueel een student en geeft aan of hij een nieuwe student wil opvoeren, een student wil verwijderen of gegevens van een student wil wijzigen. Indien hij een student wil opvoeren wordt sub-flow 4.1 uitgevoerd, indien hij gegevens van een student wil wijzigen wordt sub-flow 4.2 uitgevoerd en bij verwijderen wordt sub-flow 4.3 uitgevoerd
4.1.1	Het systeem toont een leeg invoer formulier.
4.1.2	De gebruiker vult de gegevens in
4.1.3	Indien de gebruiker behaalde cijfers wil toevoegen wordt UC01.01 uitgevoerd
4.1.4	De gebruiker geeft aan dat de gegevens kunnen worden opgeslagen, waarna het systeem de gegevens verwerkt in de database.
4.2.1	Het systeem toont de gegevens van de geselecteerde student die vervolgens door de gebruiker worden aangepast. Voor het beheren van vakken wordt UC01.01 uitgevoerd
4.2.2	De gebruiker geeft aan dat de gegevens kunnen worden opgeslagen, waarna het systeem de gegevens verwerkt in de database.
4.3.1	Het systeem toont de gegevens van de geselecteerde student.
4.3.2	De gebruiker geeft aan dat de gegevens kunnen worden verwijderd, waarop het systeem de gegevens verwijdert, inclusief de cijfers van de student.
5	Het proces gaat terug naar stap 2, totdat de gebruiker aangeeft dat hij wil stoppen, waarna de Use-case beëindigd wordt. Alle gegevens zijn op een correcte manier in de database verwerkt
Alternative Flows:	
6	Afbreken van de bewerking. De gebruiker heeft op ieder moment in de flow de mogelijkheid het muteren af te breken. Eventuele veranderingen in de database worden terug gedraaid. En de use-case wordt beëindigd.
4.1.1a	Bij het opvoeren van een nieuwe student wordt een bestaand studentnummer ingegeven.
4.1.1b	Het systeem geeft een foutmelding, waarna de use-case verder gaat met 4.1.2
Precondities:	Gebruiker is ingelogd en beschikt over de rechten om studentgegevens te muteren.
Postcondities:	Studentgegevens van betreffende student zijn gewijzigd
Business Rules	Een studentnummer mag niet gewijzigd worden
Supplementary Requirements:	Beheer moet plaats kunnen vinden 24uur/dag, 7 dagen/week.
Opmerkingen:	Woonplaats gegevens moeten kunnen worden geselecteerd uit een lijst Gerelateerde gegevens: Studentnummer, Studentnaam, Woonadres, Postadres, Telefoon, Behaalde vakken, cijfers
Openstaande punten:	Datum waarop een cijfer behaald is wordt niet vastgelegd. Hier geldt echter wel een verjaringstermijn. Uitzoeken of het noodzakelijk is om dit gegeven vast te leggen.

Basis stroom

De beschrijving van de use-case is opgebouwd uit stappen. De stappen van de use-case beschrijven de sequentie van acties die door het systeem of actoren worden uitgevoerd. In deze beschrijving wordt ervan uitgegaan dat er geen fouten in de acties optreden. Ook kunnen in deze sequentie van acties geen conditionele stappen worden geplaatst. Het is dus een typisch scenario dat altijd correct verloopt. Fout scenario's moeten op min of meer dezelfde wijze worden beschreven, maar dan als alternatieve stromen.

T.a.v. de individuele stappen van de een stroom gelden de volgende richtlijnen:

Stappen dienen te worden genummerd om de volgorde van de sequentie te verduidelijken

Wanneer bepaalde stappen moeten worden herhaald, moet dit expliciet worden opgenomen onder de laatste stap van de herhaling.

Wanneer in een bepaalde stap een andere (sub) use-case wordt aangeroepen (een include) dient dit expliciet te worden aangegeven.

Geef bij de laatste stap expliciet aan dat dit het einde van de use-case betreft.

Alternatieve stromen

Voor alternatieve stromen gelden dezelfde regels als gesteld bij de sectie "Basis stroom". Het vastleggen van fout scenario's middels alternatieven is van belang voor het vastleggen van belangrijke business rules. Er kunnen zich twee soorten alternatieven voordoen:

- **Excepties:** In het geval van excepties beschrijft het alternatief de condities zoals die kunnen voorkomen in de verschillende acties (stappen) en welke andere acties of use-cases die tot gevolg hebben. Een voorbeeld van een exceptie is de alternatieve pad dat volgt indien iemand een verkeerd wachtwoord invoert bij het inloggen.
- **Variaties:** In het geval van variaties wordt geen conditionele stap uitgevoerd. De variatie zegt iets over de verschillende varianten die kunnen voorkomen bij een betreffende stap. Vaak is dit een alternatieve actor of een technologisch alternatief. Een voorbeeld van een variatie: als iemand van de examencommissie inlogt om cijfers te bekijken, moet die eerst selecteren van welk student hij de cijfers wil zien, terwijl bij de student die keuze niet bestaat (kan alleen zijn eigen cijfers zien).

Elke alternatieve stroom heeft unieke identificatie voor de traceerbaarheid. In elke alternatieve stroom wordt gerefereerd naar de stap (actie) waarop het alternatief betrekking heeft. De stappen die uitgevoerd worden als de conditie geldig is (TRUE) worden vervolgens genummerd. Geef bij de laatste stap van de alternatieve stroom expliciet aan hoe de use-case vervolgd wordt.

Use-case diagram

- Use-case generalisaties moeten fysiek boven use-case specialisaties worden geplaatst;
- Indien een extensie de vorm heeft van een sub use-case moeten extensies fysiek onder de basis use-case worden geplaatst;
- Indien de overzichtelijkheid dit vereist kunnen actors en use-cases met een logische samenhang worden gegroepeerd middels use-case packages.

Associaties

Use-cases en Actors kunnen signalen naar elkaar sturen. In use-case diagrammen worden deze signalen weergegeven door middel van associaties. T.a.v. associaties gelden de volgende richtlijnen:

- Om de leesbaarheid van het use-case diagram te bevorderen kan tussen een use-case en een actor maximaal 1 associatie voorkomen (ongeacht het aantal transmissies dat tussen de actor en de use-case plaats vinden)
- Aangezien een associatie meerdere transmissies kan beslaan, worden associaties niet van een naam voorzien

- In het use-case diagram wordt de initiërende (primaire) actor aangeduid met een uitgaande pijl naar de use-case.
NB. Pijlen om de richting van communicatie aan te duiden worden derhalve niet gebruikt.

Use-case generalisatie

Het gebruik van use-case generalisatie dient te worden vermeden maar is wel toegestaan. Wanneer twee use-cases slechts weinig van elkaar verschillen is het beter om er één use-case van te maken waarin de afwijkende stappen als alternatieve stromen (variaties) worden uitgewerkt.

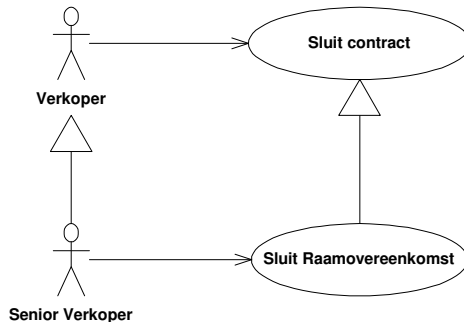
Actor generalisatie

Het gebruik van actor generalisatie moet worden vermeden maar is wel toegestaan. Als gebruik wordt gemaakt van actor generalisatie moeten de volgende uitgangspunten in acht worden genomen:

1. Een gespecialiseerde actor mag geen gespecialiseerde use-case uitvoeren.
2. Een gespecialiseerde actor kan alle use-cases van de gegeneraliseerde actor uitvoeren.

1. Een gespecialiseerde actor mag geen gespecialiseerde use-case uitvoeren

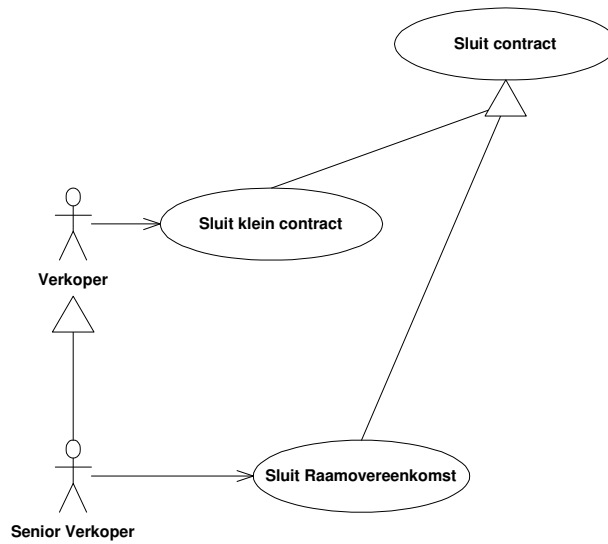
De motivatie voor uitgangspunt 1 kan worden gegeven aan de hand van het volgende voorbeeld:



Figuur 1 - Foutief gebruik actor generalisatie

In UML is gesteld dat een use-case specialisatie altijd zijn bovenliggende use-case generalisatie kan vervangen. In het bovenstaande voorbeeld kan de actor verkoper dus ook een raamovereenkomst sluiten terwijl hij er hier, organisatorisch gezien, geen bevoegdheid toe heeft.

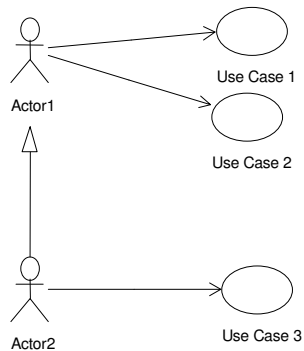
In het onderstaande figuur is aangegeven hoe dit probleem moet worden opgelost.



Figuur 2 - Correct gebruik actor generalisatie

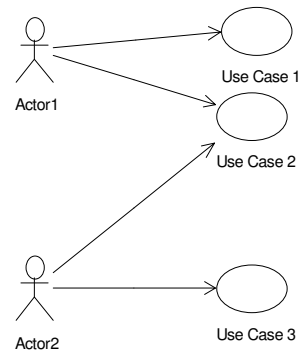
2. Een gespecialiseerde actor kan alle use-cases van een gegeneraliseerde actor uitvoeren

Wanneer de actor slechts een subset van de use-case van een andere actor mag uitvoeren, kan actor specialisatie niet worden toegepast.



Figuur 3 – Met actor generalisatie

- Actor1 kan use-cases 1 en 2 uitvoeren
- Actor2 kan use-cases 1,2 en 3 uitvoeren



Figuur 4 - Zonder actor generalisatie

- Actor1 kan use-cases 1 en 2 uitvoeren
- Actor2 kan use-cases 2 en 3 uitvoeren

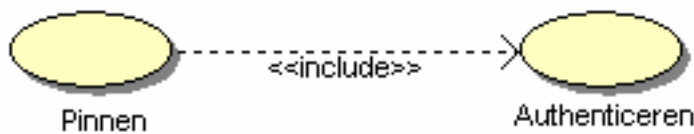
Include relatie

Include relatie wordt gebruikt als een use-case in zijn geheel gebruik maakt van een andere use-case. Include relaties zijn maximaal één niveau diep. Include relaties kunnen worden gebruikt om een use-case model te (her)structureren maar worden doorgaans niet opgenomen in een initieel use-case model.

Gebruik include relaties voor de volgende zaken:

- Scheiden van functionaliteit van de basis use-case die niet noodzakelijk is voor het begrijpen van de primaire doelstelling, maar waarvan wel het resultaat belangrijk is.
- Scheiden van functionaliteit die kan worden gebruikt in twee of meer use-cases;

Een voorbeeld. Om geld te kunnen pinnen wordt gebruik gemaakt van de use-case:



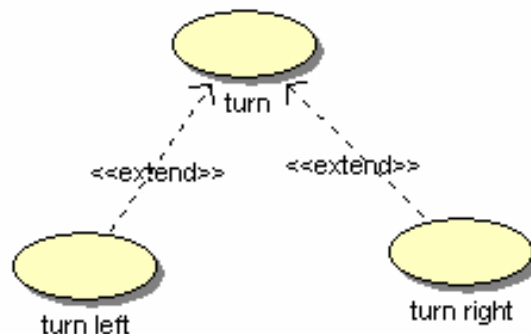
Extend relatie

Extend relatie wordt gebruikt als een use-case een gecontroleerde variatie (uitbreiding) op een bestaande use-case is. Extend relaties zijn maximaal één niveau diep. Extend relaties hebben als primair doel het aanbrengen van alternatieve stappen of scenario's zonder de base use-case te hoeven aanpassen. Extend relaties worden doorgaans niet opgenomen in een initieel use-case model.

Gebruik extend relaties in de volgende gevallen:

- Het normale verloop van een basis use-case kan worden onderbroken (interrupt);
- Een basis use-case is een trigger voor een asynchroon proces dat het verloop van de basis use-case niet verstoort;

Een voorbeeld:



Supplementary specifications

Er is nog een document behorende bij een use-case waar we in deze cursus geen aandacht aan zullen besteden, maar waarvan je wel moet weten dat hij bestaat, het "Supplementary specifications" document.

In het "Supplementary specifications" document worden alle niet-functionele requirements beschreven die niet reeds als special requirement bij de use-case specifications zijn opgenomen.

Denk hierbij bijvoorbeeld aan:

- Beschikbaarheid *de tijden waarop de applicatie beschikbaar moet zijn*
- Security *expliciete eisen t.a.v. de beveiliging van de gegevens*
- Performance *algemene eisen t.a.v. de performance van de applicatie*
- Platformen *eisen t.a.v. de ondersteuning van meerdere platformen*
- Helpfunctionaliteit *het al dan niet beschikbaar zijn van online helpfunctionaliteit*

Aanvullende opmerkingen

- Gebruik zo eenvoudig mogelijk zinnen (Onderwerp, werkwoord, lijdend voorwerk, bijzin) en zorg dat duidelijk is wie de stap bestuurd, bv. 'Het systeem toont het rekeningsaldo', of 'De klant geeft zijn pincode in'.
- Schrijft **altijd** vanuit de gebruiker. Bv 'De klant voert pincode in', en niet 'Lees Pincode'.
- De Actor moet specifiek in de tekst van de diverse flows benoemd worden; "Actor" of de naam van de Actor. Het gebruik van de kreet "Gebruiker" zoveel mogelijk vermijden.
- Een use-case wordt meestal door één specifieke trigger gestart. Een use-case die meerdere triggers heeft zou dus eigenlijk gesplitst moeten worden.
- De beschrijving van de interne werking van een systeem hoort niet thuis in een use-case
- Create, Read, Update en Delete (CRUD) resulteert in principe tot één use-case. Hiervan kan afgeweken worden als de werking sterk afwijkt en de beschrijving van de use-case te complex wordt. Met andere woorden; als de business erom vraagt. Bijvoorbeeld: complexe autorisatie mechanismen bij raadpleegfuncties.
- Maak de stappen niet te gedetailleerd. Zorg voor maximaal 9-10 stappen in een flow-of-events. Bij meer worden de stappen waarschijnlijk te gedetailleerd, bij minder te grof. Indien te gedetailleerd, helpt het om te vragen waarom de actor dit doet, met welk doel. Hiermee kunnen stappen vaak gecombineerd worden.
- Zorg dat een use-case de bedoeling van de actor toont, niet zijn handelingen. Beschrijving van de gebruikshandleiding is een veel voorkomende fout. In feite beschrijf je dan de user-interface dialoog. Hierdoor worde de use-case slecht leesbaar en meer onderhoudsgevoelig. Een voorbeeld: 'Systeem vraagt naam, Gebruiker geeft naam, Systeem vraagt om adres, Gebruiker voert adres in, gebruiker klik op OK, het systeem toont de gebruiker-profiel', beter is: 'De gebruiker voert zijn naam en adres in, Het systeem toont de gebruiker-profiel'.
- Stop de use-case niet vol met allemaal beschrijvingen over de te gebruiken attributen. Dit hoort thuis in de klassen.
- Een use-case *controleert* niet, maar *valideert*. Een mogelijk actie-stap kan zijn het valideren van een business regel (bv, ingevoerde pincode). Vaak schrijft men dat de regel

gecontroleerd wordt. Dit is niet goed, controle is niet het echte doel en het resultaat wordt ook niet duidelijk tenzij de controle direct gevolgd wordt door "Indien goed, dan.." Beter is: "Het systeem valideert dat wachtwoord correct is." Een mogelijk foutscenario (wachtwoord incorrect) vinden we terug bij de alternatieve flows.

- Timing is optioneel, stappen worden in volgorde uitgevoerd.
- Communicatie met andere systemen, altijd gestuurd door de actor. Stel dat het systeem informatie van een ander systeem moet ophalen, dan kunnen we niet schrijven: 'Gebruiker klikt op de 'ophaal' knop, waarna het systeem de informatie ophaalt.' Beter is: 'Gebruiker laat het systeem informatie ophalen van systeem B'
- Stappen kunnen herhaald worden, tot een conditie is bereikt. Herhalen van stappen kan zonder meer. Indien meerdere stappen herhaald moeten worden kunnen we dit aan het begin of het einde van de stappen aangeven.
- Een actor kan een systeem zijn. Wees hier wel voorzichtig mee. Vraag je altijd af of het echt het systeem zelf is die de rol uitvoert of toch een persoon met een bepaalde rol.
- Een open deur, maar toch: de use-case beschrijft het WAT, niet het HOE.