

H02: Functionele requirements

1. Inleiding

Zoals gezien tijdens het opleidingsonderdeel Software Analysis zijn requirements of behoeften de eisen van de klant waaraan een product moet voldoen. Er werd toen reeds een onderscheid gemaakt tussen functionele requirements en niet-functionele requirements. Deze functionele behoeften vertellen welke functionaliteiten het systeem moet bevatten en die dus door een actor moeten kunnen uitgevoerd worden op het systeem. Deze functionaliteiten zullen we vastleggen en éénduidig definiëren in use cases.

2. Use case diagram

Een overzicht van alle beschikbare functionaliteiten wordt dan ook weergegeven in een use case diagram. Enerzijds hebben we in dat schema een overzicht van alle rollen die toegang hebben tot het systeem. Daarnaast toont een use case diagram ook welke rol gebruik kan maken van welke functionaliteit op het systeem. Het kan echter zijn dat voor het uitvoeren van een bepaalde functionaliteit steeds een andere nodig is, in dat geval gebruiken we het mechanisme van “include”. Het schema kan ook gebruiken van een “extends” tussen twee use cases indien de ene functionaliteit soms nood heeft aan een andere bij het uitvoeren.

3. Use cases

3.1. Inleiding

Een use case bevat het uitgeschreven verhaal van een primary actor die het systeem zal gebruiken om een bepaald doel te bereiken. Deze verhalen zijn in een zo eenvoudig mogelijke taal opgesteld, dus geen gebruik van technisch jargon, met als belangrijkste doel: duidelijke communicatie. Enerzijds is er communicatie naar de klant toe. We kunnen dus ons uitgeschreven verhaal aan de klant voorleggen om te verifiëren of we effectief het verhaal beschreven hebben zoals hij het bedoelt heeft. Anderzijds dient een use case om te communiceren met de mensen uit het IT team om het product verder te ontwikkelen. Een use case is een visie op het systeem met een bepaalde scope, afgebakend doel, waarin bepaalde belanghebbende partijen in interactie treden met het systeem om dat welbepaalde doel te bereiken. Deze verhalen zijn ook kort van aard, gezien het als duidelijk communicatiemiddel moet dienen, en worden dan ook per scenario beperkt tot een halve pagina.

Hoe weten we nu welke verhalen we gaan uitwerken als use case? Daarvoor gaan we op zoek naar de elementaire business processen die een user goal voorstellen.

3.2. Elementair business proces

Elementaire business processen zijn op te delen in drie verschillende categorieën. Enerzijds hebben we hogere doelen van een bedrijf die een verzameling zijn van aparte doelen. Dit noemen we “summary goals”

3.3. Onderdelen

Een use case bevat volgende elementen/onderdelen:

- Primary actor
- Stakeholder(s)

- Preconditie(s)
- Postconditie(s)
- Normaal verloop
- Alternatieve verlopen
- Domeinspecifieke regels
- Op te klaren punten (optioneel)

3.3.1. Primary actor & stakeholders

De primaire acteur is de persoon die het verhaaltje activeert. Hij is dus de persoon die de functionaliteit wenst uit te voeren op het systeem.

Een stakeholder of belanghebbende is een persoon die er belang bij heeft dat primary actor zijn doel kan realiseren maar zal zelf niet actief deelnemen aan het verhaal.

3.3.2. Precondities & postcondities

In de precondities vermeldde we de niet triviale voorwaarden die moeten voldaan zijn voor een use case kan starten. Dit zijn dus voorwaarden die door het systeem kunnen gevalideerd worden voor het verhaal start. Het is bijgevolg dus niet meer nodig die nadien nog af te toetsen tijdens een scenario van de use case. In veel gevallen zal je hier een verwijzing naar een externe use case tegenkomen.

Bij de postcondities geef je vervolgens aan wat er door het systeem vervuld is na het uitvoeren van een scenario. Hierin staan dus de bereikte wijzigingen van het systeem t.o.v. de status van het domeinmodel voor het starten van de use case. Niet elk alternatief scenario zal deze postcondities bereiken.

3.3.3. Normaal verloop & alternatief verloop

Het normale verloop van een use case beschrijft de main success story. Dit is dus een uitgeschreven stappenplan van hoe de use case het meeste wordt gebruikt. Het is een chronologische en oplopend genummerde volgorde van actiestappen waarbij een wisselwerking tussen de primaire actor en het systeem wordt beschreven. In één stap is het onmogelijk dat de primary actor EN het systeem iets doen! Vermijdt het uitschrijven van UI acties. We gaan er altijd van uit dat de primary actor de stappen wil uitvoeren, dus bevestigingen en annulaties zijn overbodig tenzij deze toch een speciaal verloop zouden hebben!

Mogelijke stappen zijn:

- Primary actor voert actie uit
- Systeem registreert interne wijziging
- Systeem voert validatie of actie uit
- Systeem vraagt naar gegevens
- Systeem toont een melding en/of stuurt een melding

Bij de alternatieve verlopen worden de verhalen beschreven wanneer een stap uit het normale verloop of een stap uit een ander alternatief verloop afwijkt van het verhaal.

Elk alternatief verloop heeft als nummering de stap van het oorspronkelijke scenario waar de afwijking start aangevuld met een alfabetische nummering van alternatieve verlopen voor die stap.

Binnen het alternatief verloop zelf wordt dan de nummering aangevuld met een oplopend nummer (start opnieuw bij 1) van chronologisch opeenvolgende deelstappen.

Een alternatief verloop kan op volgende manieren eindigen:

- Verhaal keert terug naar een stap in het normaal verloop
- Verhaal keert terug naar een stap in een ander alternatief verloop
- Er wordt een externe use case opgeroepen
- De huidige use case wordt stopgezet. Vermeldt hierbij steeds de bereikte postcondities

3.3.4. Domeinspecifieke regels & op te klaren punten

Dit stuk bevat een opsomming van alle relevante regels die te maken hebben met het domein van het probleem. Hier vermeld je dus ook alle technische gegevens die later nodig zijn voor de ontwerper, ontwikkelaar of tester. B.v.: Het wachtwoord moet minstens 8 tekens lang zijn en minstens een hoofdletter bevatten. De naamgeving per regel is steeds DR_<naam>. De <naam> is vrij te kiezen maar zorg voor een passende en duidelijke naam.

Onder op te klaren punten worden alle zaken die nog onduidelijk waren bij het opstellen van de use case opgesomd. Deze sectie is vooral bedoelt om te gaan communiceren met de klant. Er is met andere woorden meer nood aan uitleg van de klant/opdrachtgever. Op het einde van de analysefase tijdens het software-ontwikkelingsproces is deze sectie dan ook leeg en overbodig geworden want je zou alle info van de klant moeten hebben verkregen op dat moment.