## Reference Card

## Use Cases en Activity Diagrams

**Zoals bij natuurlijke taal komen er ook in de UML veel dialecten voor. In zo’n dialect zijn er net iets andere afspraken, aangepaste layout of andere naamgeving van toepassing. De UML artifacten van bedrijven, organisaties e.d. zijn dan ook vaak verschillend maar met enige moeite voor ons wel te begrijpen.**

**Omwille van de leesbaarheid en de eenduidigheid spreken we daarom binnen SAQ een aantal zaken af in deze Reference Card van de ICA standaard.**

**Use Cases:**

* De naam van een use case bevat een werkwoord. En is uniek en duidelijk. De naam is betekenisvol voor de Actor.
* Formuleer alles altijd actief en niet passief, zodat duidelijk is wie welke handelingen doet. Dus niet "Het *wordt* ingevuld", maar "De *actor* vult dit in: ..".
* We nummeren nooit dieper dan drie niveaus in een stappenplan. Als dit nodig blijkt, klopt de complexiteit van de use case niet en moet deze gesplitst worden.
* Meerdere pre- of postcondities scheiden we altijd met het woord *'of'* of *'en'*.
* De pre/postcondities zijn relevant voor het systeem. Dus geen dingen als "Er moet een printer aanwezig zijn".
* Het einde van een use case geef je aan met het woord "Stop". Dit kan na een (niet geslaagde) voorwaarde zijn of aan het einde van de use case, maar dit laatste is triviaal dus mag weggelaten worden.
* Geen Crud Use Cases (dus **geen** implementatie keuzes maken).
* In de Use Case Model spreken we van een of meerdere Actors en niet van Acteurs.
* Voor het aanroepen van een secundaire use case gebruiken we:   
  "Voer uit **Invoeren NAW**", waarbij 'Invoeren NAW' een secundaire use case is.

**Extended Use Cases (ook wel: Use Case Outline):**

* Stappenplan of Outline van een Use Case noemen we Extended Use Case.
* Het stappenplan in een use case beschrijving is altijd genummerd.
* We onderscheiden apart de Basic Flow en de Alternative Flow
* Ook de Alternative flows, uitzonderingen en faalscenario’s beschrijven we (indien mogelijk) in 2 kolommen. Maar duidelijk waar deze flows “inhaken” op de Basic Flow.
* We geven links in een tabel de acties van de gebruiker en rechts de acties van het systeem.
* De Basic Flow start altijd met een actie van de gebruiker en eindigt met een actie van het systeem.
* Zijn actief geschreven, zodat duidelijk is wie welke handelingen doet. Dus niet "Het *wordt* ingevoerd", maar "De *actor* voert “bla” in: ..". Vermijdt dus de werkwoorden worden en zijn.
* Trigger moet beschreven zijn in de stap
* Het is altijd duidelijk welke Actor de stap uitvoert.
* Alle mogelijke postcondities zijn toegevoegd.

(voorbeeld)

|  |  |
| --- | --- |
| ***Use Case: name (Unieke, herkenbare naam, overnemen uit UCD)*** | |
| ***Purpose: …..( Wat bereikt de primaire actor met de use case? Formuleer actief: “De apotheker vraagt…”)*** | |
| ***Description of use case: … (Korte ‘managements’-samenvatting van handelingen. Hoe wordt het doel bereikt?)*** | |
| ***Primary actor*: Cashier**  ***Secondary actor***: **External accounting, Inventory systems** | |
| ***Stakeholders and interests*: Cashier, Customer, Company, etc.** | |
| ***Preconditions*: Cashier is identified and authenticated**  ***(welke voorwaarden moet voldaan zijn vóór aanvang use case? Dit kunnen er meerdere zijn!)*** | |
| ***Postconditions (Success Guarantee)*: Sale is saved.Tax is correctly calculated.**  ***elke verschillende situaties worden met de use case bereikt? Dit kunnen er meerdere zijn!)*** | |
| ***Basic Flow (Main Success Scenario)*** | |
| ***Actor action*** | ***System responsibility*** |
| **1. Customer arrives at a POS checkout with goods to purchase.** |  |
| **2. Cashier starts a new sale.** |  |
| **3. Cashier enters item identifier** |  |
|  | **4. System records each sale line item and presents item description and running total. Price calculated from a set of price rules.** |
| **Cashier repeats steps 3-4 until indicates done.** |  |
|  | **5. System presents total with taxes calculated.** |
| **6. Cashier tells Customer the total, and asks for payment.** |  |
| **7. Cashier accepts payment from customer and enters amount in system.** |  |
|  | **8. System handles payment and prints receipt. System calculates Cashback.** |
| **9. Cashier gives receipt to customer.** |  |
|  | **10. System logs the completed sale and sends information to external accounting and inventory systems. System presents receipt.** |
| ***Alternative flows*:**  **10a. Customer doesn’t accept receipt. Cashier disposes receipt.** | |
| **10b. Cashier gives cash back if systems calculated change >0.** | |
|  | |

\*\*Use case is uitgewerkt voor contante betalingen.

**Eisen Activity Diagrams**

* Gebruik van de decision node is verplicht.
* Gebruik van guards is verplicht. Gebruik eventueel de else guard.
* Gebruik van merge node is verplicht
* De acties staan onder elkaar.