

Criteria

In dit document staan criteria vermeld die je kunt gebruiken om de diverse producten te beoordelen. Wat algemene opmerkingen vooraf:

1. Denk eraan dat deze criteria niet een vervanging zijn van de lesstof (slides en boek) maar een aanvulling of een verduidelijking.
2. Let bij de diverse UML-diagrammen ook op correcte syntax en semantiek: de "L" van UML staat voor Language. En talen moeten wel correct gebruikt worden.
3. Deze lijst is niet uitputtend en geen vervanging voor nadenken.....
4. Wellicht breid je deze lijst uit met eigen dingen?

Use case model

1. Alle menselijke gebruikers van het systeem vervullen tenminste 1 rol.
2. Alle handelingen met het systeem vormen (impliciet) onderdeel van tenminste 1 use case.
3. De naam van de actor geeft de rol van de actor bij de use case aan.
4. De naam van de use case bestaat uit een actief werkwoord en een lijdend voorwerp, eventueel aangevuld met een meewerkend voorwerp.
5. De naam van de actor en de naam van de use case vormen een zin.
6. Alle actoren zijn bij tenminste 1 use case primary actor.
7. Alle use cases worden door ten minste 1 primary actor uitgevoerd.
8. De omvang van de use case is zo gekozen dat deze de "boss test" of de "EBP test" doorstaat (dit geldt niet voor een included use case, wel voor een extending use case): een use case beschrijft een korte, in tijd en ruimte begrensde interactie tussen gebruiker en systeem dat leidt tot het vervullen van een meetbaar "business goal" en die het systeem in een consistente staat achterlaat.
9. De systeemgrens is aangegeven.
10. Gewone actoren staan aan de linkerkant van het systeem, supporting actoren aan de rechterkant van het systeem

Brief format use case beschrijving

1. Een brief format is idealiter twee tot drie zinnen lang.
2. Uit de beschrijving komt duidelijk naar voren dat een gebruiker met behulp van het systeem een concreet doel wil bereiken in de vorm "actor wil met behulp van systeem doel bereiken".
3. Uit de beschrijving komt duidelijk naar voren wat er gebeurt als het doel niet gehaald kan worden.

Fully dressed use case beschrijving

1. Een fully dressed use case is altijd zo compleet mogelijk.
2. De (primary) actor is degene die de use case uitvoert.
3. Een stakeholder heeft belang bij het realiseren van de postconditie maar voert geen acties in de use case uit. De stakeholder heeft een belang bij een use case als (delen van) de postconditie van de use case onderdeel is van (een deel van) een preconditie van een use case waarvan de stakeholder de primary actor is.
4. Pre- en postcondities zijn booleaanse expressies met daarin alleen termen uit het domeinmodel (concepten, attributen en associaties) die toestanden en veranderingen in het domein beschrijven:
 1. Precondities:
 1. instanties wel/niet aanwezig

2. attributen hebben wel of niet een bepaalde waarde
3. associaties wel/niet aanwezig
2. Postcondities:
 1. instanties aanmaken/verwijderen,
 2. attributen aanmaken/verwijderen/veranderen,
 3. associaties aanmaken/verwijderen/veranderen.
5. De actor begint altijd met de use case.
6. De eerste actie van de actor mag niet bestaan uit een alternative flow/extensie.
7. Het systeem geeft altijd een response/reactie op een actie van de actor.
8. Precondities zijn, of triviaal buiten het systeem waar (te maken), of er is een use case waarin deze waar gemaakt worden (onderdeel van een postconditie).
9. Voor ieder mogelijk einde van een use case (via alternative flow/extensie) is een postconditie geformuleerd.
10. Voor elke alternative flow is aangegeven bij welke stap van het main success scenario dit alternatief optreedt.

Domeinmodel

1. Alle namen voor concepten, attributen en associaties komen voor in de fully dressed use cases en andere beschrijvingen van het domein (modulo vertaling naar Engels).
2. Alle fully dressed use cases moeten geparafraseerd kunnen worden met de concepten, attributen en associaties uit domeinmodel.
3. Alle concepten, attributen en associaties die in de pre- en postcondities van de fully dressed use cases worden gebruikt komen voor in het domeinmodel en de booleaanse expressies moeten kunnen worden geëvalueerd.
4. Concepten worden niet opgenomen als attribuut (gebruik van associaties).
5. Associaties hebben tenminste rollen. Als de associatie niet een "heeft-een" betekenis heeft, dan moet de associatie ook een label hebben.
6. Labels en rolnamen zijn zo gekozen dat de betekenis van de associatie duidelijk wordt.
7. Conceptuele classes in een domeinmodel hebben geen visibility en datatypen voor attributen en ook geen methods.
8. Bij associaties is navigeerbaarheid niet aangegeven.
9. Tekstuele beschrijving van alle concepten met daarin de belangrijkste functionaliteit van het concept.

Component diagram

1. Heeft de component een tekstuele beschrijving van de functionaliteit?
2. Heeft het component tenminste 1 offered interface?
3. Voor iedere required interface bestaat een offered interface?
4. Voor iedere associatie met een component op een andere node bestaat een specificatie van het RPC-mechanisme?

Deployment diagram

1. Iedere node is een device of een execution environment?
2. Iedere node bevat artefacten?
3. De artefacten van een node zijn compatibel?
4. Ieder artefact is een fysieke entiteit?
5. Iedere node heeft tenminste 1 communicatiepad naar een andere (eventuele externe) node?
6. Zijn alle communicatiepaden voorzien een specificatie van het RPC-mechanisme?

7. Alle nodes, artefacten en communicatiepaden zijn voorzien van zinvolle stereotypes?

Interface

1. Heeft de interface een tekstuele beschrijving van de functionaliteit?
2. Is de interface minimaal (cohesion, coupling, separation of concerns)?

Voor alle diagrammen die het gedrag van objecten beschrijven gelden de volgende regels:

1. Bestaat het object waar een message naar toe gestuurd wordt?
2. Als het object bestaat, kan er wel een message verstuurd worden naar dat object:
 1. Heeft de verzender een associatie met de ontvanger?
 2. Heeft de verzender de ontvanger als attribuut?
 3. Kan de verzender de ontvanger bereiken met behulp van een functie-aanroep van een associatie of attribuut, eventueel via-via?
3. Spelen alle concepten, attributen en associaties uit de bijbehorende pre- en post-condities van de bijbehorende use case een rol in het diagram?
4. Is uit het verloop van het diagram aannemelijk te maken dat alle post-condities van de bijbehorende use case gerealiseerd worden?
5. Corresponderen de pre-condities van de bijbehorende use case met de "initial node" of start van het diagram?
6. Corresponderen de post-condities van de bijbehorende use case met de "end node" of einde van het diagram?

System Sequence diagram

1. De messages corresponderen met de acties van de user in de use case.
2. De namen van de messages geven de intentie van de message aan.
3. De namen van de messages zijn gebiedende wijs en drukken een actie uit die het systeem zal uitvoeren.
4. Het systeem is black box

Sequence diagram

1. Corresponderen de diverse messages met de user acties in de use case?
2. Geven de namen van de messages de intentie van de message aan?
3. Corresponderen de diverse messages met de functies in het class diagram (consistentie sequence/class diagram)?
4. Is iedere message door een goed informaticus te implementeren op basis van de pre- en post-conditie van de message dan wel op basis van de messagenaam?
5. Komen alle alternative flows/extensions terug in de logica van het diagram?
6. Is het diagram "lineair" genoeg?

Activity diagram

1. Bestaan de action node naam uit een actief werkwoord en een lijdend voorwerp, eventueel aangevuld met een meewerkend voorwerp?
2. Geven de namen van de action node de intentie van de action node aan?
3. Corresponderen de user acties in de use case met action nodes?
4. Corresponderen de action nodes met de functies in het class diagram (consistentie activity/class diagram)?
5. Is iedere action node door een goed informaticus te implementeren op basis van de

- pre- en post-conditie van de action node dan wel op basis van de messagenaam?
6. Komen alle alternative flows/extensions terug in de logica van het diagram?
 7. Hebben alle uitgangen van beslissingsnodes een guard en is er logisch altijd tenminste en ten hoogste 1 guard true?
 8. Eindigen alle flows in een "flow final"?
 9. Als er "activity finals" worden gebruikt: beïnvloedt (verhindert) het bereiken van een "activity final" het bereiken van enig post-conditie?