Samenvatting Software Analysis

# Introduction

Een stakeholder is een persoon, groep of organisatie die interesse of bezorgdheid heeft over het project. Stakeholders hebben of kunnen beïnvloed worden door de acties, doelen en policies van het project.

Een projectleiden moet rekening houden met de scope, functionaliteit, budget, resources, tijd, en kwaliteit.

## Requirements

Een requirement is er om een probleem volledig en ondubbelzinnig te hebben die geen vakjargon gebruikt, en dus verstaanbaar is voor de klant. Een requirement is ook een conditie of mogelijkheid genoodzaakt door een gebruiker of stakeholder om een probleem op te lossen of een doel te halen.

De scope of het bereik van de requirements zijn de som van producten, services en resultaten die het project moet kunnen garanderen. De scope van een project is het werk dat gedaan moet worden om een product, service of resultaat met de gespecifieerde features en functies te leveren. De scope van een product zijn de features en de functies dat het product, service of resultaat definiëren.

### Soorten

#### Functionele requirement

Definieert een functie dat door het systeem moet geleverd worden, dat gemaakt moet worden.

#### Quality requirement

Definieert een kwalitatieve eigenschap dat door het systeem moet gehaald worden, dat gemaakt moet worden of 1 van de functies moet hebben.

#### Constraint

Dit is een requirement dat het project beperkt en er voor zorgt het alleen het gene dat nodig en nuttig is voor de andere requirements te halen, beschikbaar wordt gesteld.

## Woordenschat

|  |  |
| --- | --- |
| Functionele requirement | Een beschrijving van het gedrag dat een systeem zal tonen onder specifieke omstandigheden. |
| Constraint | Een beperking die ingesteld is voor de ontwikkelaar omtrent design en constructie van het product. |
| Business requirement | Een ‘high-level’ businessdoel van de organisatie of de klant wie het product aanschaft. |
| Systeem requirement | Een ‘top-level’ requirement van een product dat meerdere subsystemen heeft, welke allemaal software of software en hardware kunnen zijn. |
| Quality attribute of kwaliteitseigenschap | Een soort van niet-functionele requirement dat een service of prestatie-eigenschap van het systeem beschrijft. |
| External interface requirement | Een beschrijving van een connectie tussen het softwaresysteem en de gebruiker, een ander software systeem of een hardware-apparaat. |
| Niet-functionele requirement | Een beschrijving van een eigenschap dat een systeem moet kunnen tonen of een beperking dat het moet respecteren. |
| Business regel | Een richtlijn, standaard of regulatie die een aspect van de business definieert of beperkt. Het is geen software-requirement op zich, maar de oorsprong van verschillende types van software-requirements. |
| User requirement | Een doel of taak dat zegt dat specifieke klassen of gebruikers met het systeem of een producteigenschap moet kunnen interacteren. |
| Feature | Een of meer logisch gerelateerde systeemmogelijkheden die van waarde betekenen voor een gebruiker en beschreven zijn door een som van functionele requirements. |

High-level:

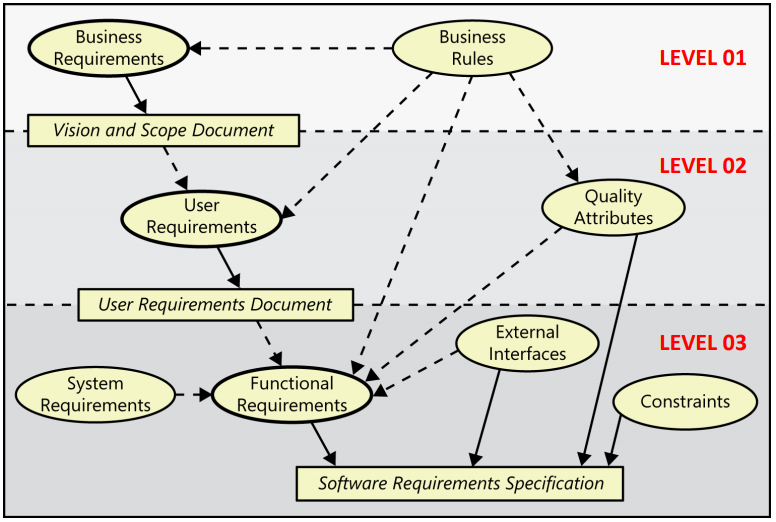
Dit zijn business requirements die het oogpunt van de sponsor benadrukken, het bereik van het product definiëren en de businessdoelen weergeeft.

Mid-level:

Dit zijn gebruikersrequirements met het oogpunt van de gebruiker benadrukke, de doelen van de gebruiker en de inputs en outputs van de gebruiker.

Detailed-level:

Dit zijn systeemrequirments die functioneel (wat het systeem doet) en non-functioneel (hoe goed het systeem het doet) zijn.



## ISO 25010

ISO 25010 is een internationale standaard voor de kwaliteit van producten:

* Portability
* Efficiency
* Reliability
* Funtional suitability
* Usability
* Maintainability
* Security

Men kan ook SMART requirements opstellen:

* Specific een requirement moet duidelijk zeggen wat gevraagd id
* Measurable een requirement moet bevestigd zijn wanneer de eisen gevolg zijn
* Attainable een requirement moet mogelijk of technisch mogelijk zijn
* Realistic een requirement moet redelijk en realistisch zijn
* Timely een requirement moet haalbaar zijn binnen een acceptabele tijdslimiet

Een theme duurt jaren, een epic duurt maanden, een feature duurt weken, een story duurt dagen en een task duurt uren.

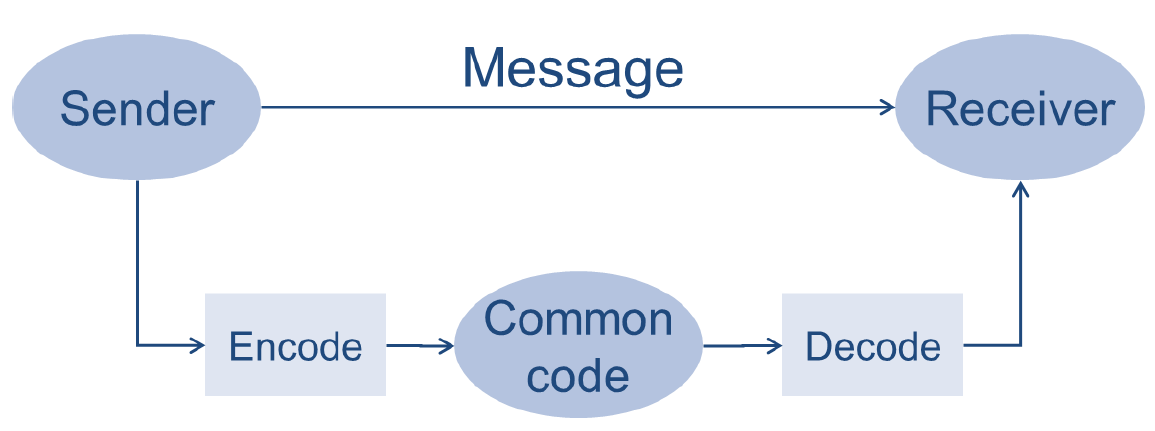
Het stappenplan van requirements engineering is in volgorde:

* System & system context
* Elicitation
* Analysis
* Specification
* Verification & Validation
* Manage & Trace (RMS)

Een goede aanpak voor specificatie en management is relevante requirements weten, een consensus kunnen halen, alles documenteren, de noden van stakeholders begrijpen en het risico kunnen minimaliseren.

De symtomen voor een slechte RE zijn onduidelijke en ontbrekende requirements. De redenen zijn slechte aannemingen, communicatieproblemen en druk van de klant.

Het standaard communicatiemodel ziet er als volgt uit:

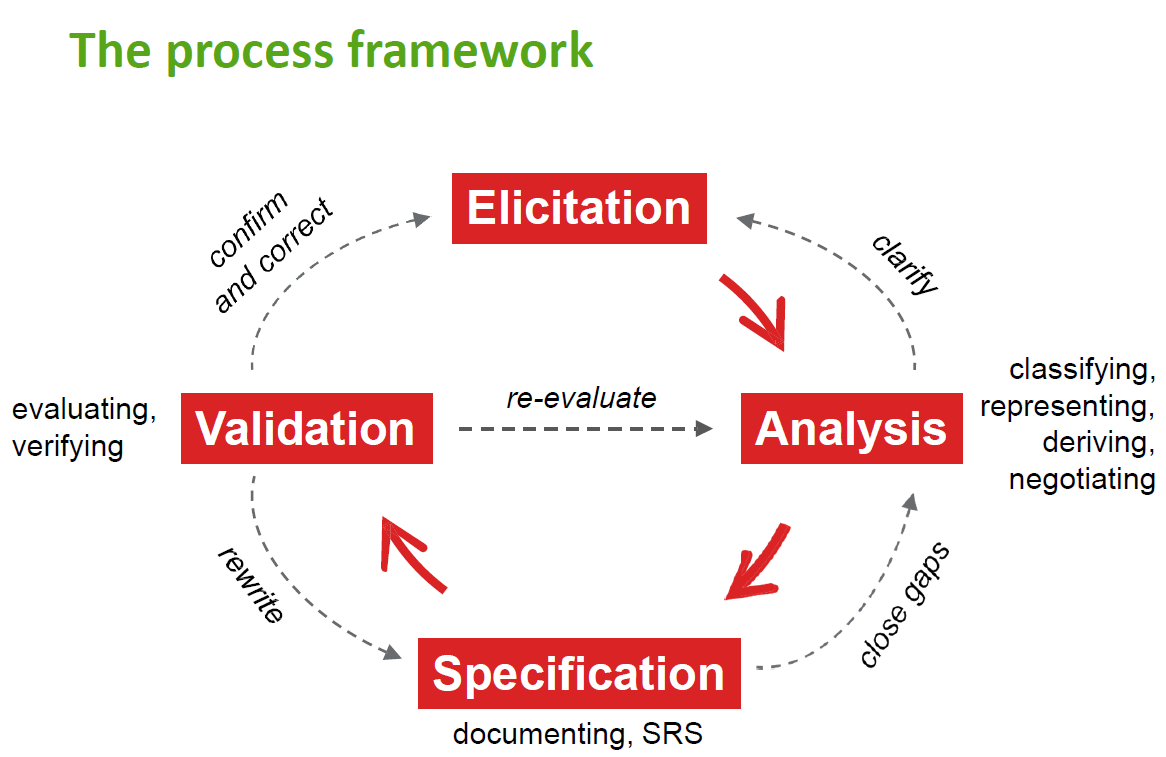


Bij communicatie is de boodschap 20% bepaald door wat er is gezegd, 30% hoe het is gezegd en 50% door de expressies.

Enkele soorten:

* Enterprise-Scale Adaptive Processen Lean & Agile
* Adaptive Processen OpenUp, Scrum, Kanban…
* Iterative Processen RAD, RUP & Spiral
* Predictive Processen Waterfall model & V-model

# System & System Context – Context Diagram



De Atlassian toolsuite omgeeft zich in Analysis en Specification van Requirements Development.

Een stakeholder is iemand die iets te winnen of te verliezen heeft bij een project. Goals bepalen het succescriteria van een project. De scope zijn de grenzen van het project.

Een SRS template in IEEE 830 ziet er zo uit:

* Introductie (Purpose & Product Scope)
* Algemene beschrijving
* Systeem features
* Data requirements
* Externe Interface Requirements
* Kwaliteitsattributen
* Internationalisatie en localisatie requirements
* Andere requirements (Glossary & Analysis Models)

Een context diagram toont welke data tussen de buitenwereld en het systeem bestaat. Dit diagram zou ook de systeemgrenzen moeten beschrijven. Dit is geen flowchart maar een zero level data flow diagram. Er bevinden zich geen data stores in dit diagram.

Fysiek DFD: model van het huidige systeem, getekend en geanalyseerd.  
Logisch DFD: model van het voorgestelde systeem, getekend van het fysieke DFD.

Een externe identiteit is een iets dat data geeft en accepteerd. Een data flow is een beweging van data. Een proces is een actie gemaakt door invoerdata naar uitvoerdata.

Een DFD heeft enkele regels:

1. Pijlen mogen elkaar niet doorkruisen
2. Vierkanten, cirkels en bestanden moeten een naam hebben
3. Opgesplitst data flows moeten gebalanceerd zijn (op origineel lijken)
4. Geen data flow, vierkant of cirkel kan dezelfde naam hebben
5. Alle data flows moeten buiten het diagram getekend worden
6. Gebruik betekenisvolle namen
7. Controle informatie (tellers, wachtwoorden en validaties) zijn niet toegelaten in een DFD

# Business Use Case

## UML – Unified Modeling Language

UML is geen methode maar een grafische presentatie van bepaalde aspecten van het systeem. Het doel van een UML is om business modeling te ondersteunen. UML heeft geen specifieke notatie voor business of business actors. BUCs worden geïntroduceerd in RUP. Voor simpele business situaties gebruiken we een contect diagram maar wanneer dit niet voldoende is, gebruiken we een BUC model.

Een BUC is een samenstelling waarbij de klant of de betrokken partijen al hun nodige activiteiten samen kunnen voegen in 1 locatie.

Een SUC is een samenstelling waarbij een gebruiker of een computer gebruik van maken van het systeem.

En zijn enkele BUC symbolen:

|  |  |  |
| --- | --- | --- |
| Symbool | Naam | Betekenis |
|  | Business Use Case | Beschrijft een proces dat waarde heeft tot een actor en beschrijft ook wat het proces doet. |
|  | Business Actor | Beschrijft een rol (klant, verkoper, etc..) die interacteert met de omgeving. |
|  | Business Boundary, Subject | Een onderwerp van een use case inclusief zijn grenzen. |
|  | Association | Een relatie tussen 2 andere objecten. |
|  | Generalization | Een relatie waarbij 1 element het kind is en gebasseerd op is de ouder. Het kind erft alle eigenschappen, operaties en relaties van de ouder. |
|  | Include | A kan niet zonder B en B weet niet wie A is. |
|  | Extend | A kan bestaan zonder B en weet niet of B bestaat. B weet wie A is. |

Multipliciteiten:

|  |  |
| --- | --- |
| Symbool | Betekenis |
| 0..1 | 0 of 1 |
| 1 | Exact 1 |
| 0..\* | 0 of meer |
| \* | 0 of meer |
| 1..\* | 1 of meer |
| 1..6 | 1 tot 6 |
| 1..3,7..10,15,19..\* | 1 tot 3 of 7 tot 10 of exact 15 of 19 of meer |

Stappenplan:

1. Vind alle actoren
2. Vind alle goalen
3. Beschrijf de pre-condities
4. Beschrijf de post-condities
5. Beschrijf de main flow
6. Beschrijf de uitzonderingen
7. Beschrijf de alternatieve flows