

Disclaimer | De lay-out van de PDF-versie voor dit document kan verschillen met de markdown versie, voor een accurate weergave zie markdown bestand in bitbucket.

Definition of Done

Natuurlijke taal

De gebruikte natuurlijke taal is in de gehele implementatie Engels. Dit geldt dus ook voor comments, functienamen, variabelenamen en commit messages. Alle documentatie (denk aan het FO en TO) wordt wel in het Nederlands geschreven.

Code

Voor alle SQL code die geschreven wordt tijdens de realisatie van de database van de applicatie wordt de SQL style guide van GitLab aangehouden (style guide, GitLab team), met een paar uitzonderingen:

- Er wordt geïndent met tabs;
- Er worden ge-aliasde data types gebruikt (zoals `NUMERIC` i.p.v. `NUMBER`), zodat dit gelijk is met PowerDesigner.

Unit-tests

Alle geschreven unit-tests voldoen aan het opgestelde testplan (testplan, GroepA4).

Functioneel Ontwerp

Het functioneel ontwerp moet voldoen aan een aantal punten:

- Alle use-cases zijn door de opdrachtgever goedgekeurd en zijn relevant aan het project;
- Alle use-cases omschrijven één actie, op één moment, door één persoon.
- Er staat **geen** implementatie-specifieke taal in de use-cases.
- Alle feittypen sluiten aan op het domein van de opdrachtgever en de bijbehorende verwoording(en);
- Bij alle feittypen wordt zo veel mogelijk informatie weggelaten, mits er geen relevante informatie verloren gaat.
- Wanneer twee of meer entiteiten van de feittypen een relatie tussen elkaar hebben, wordt deze aangegeven met een bijpassende naam;
- Stel businessrules zo specifiek mogelijk op, voorkom onduidelijkheid.
- Wanneer een business rule wordt aangemaakt krijgt deze een nummer(b.v. BR12), zodat er makkelijk naar de Business Rule verwezen kan worden;
- Voor elke business rule die niet uit het CDM kan worden afgeleid, is een constraint opgesteld;
- Alle functionele requirements zijn geprioriteerd met MoSCoW;
- Functionele requirements zijn SMART opgesteld.
- Alle ontwerpkeuzes zijn behandeld in het functioneel ontwerp. Per ontwerpkeuze wordt er ingegaan op het probleem, welke mogelijke oplossingen er voor waren met hun voor- en nadelen en voor welke oplossing uiteindelijk is gekozen en waarom;
- In de rechtenstructuur zijn alle use-cases behandeld. Per use-case is er aangegeven van welke entiteit(en) deze use-case gebruik maakt en op welke manier (**Create**, **Read**, **Update** of **Insert**). Ook wordt er aangegeven wie dit mag (data-analist en/of admin).

Technisch ontwerp

Het technisch ontwerp moet voldoen aan een aantal punten:

- Non-functional requirements zijn SMART-omschreven
- Alle elementen in het deployment diagram zijn omschreven in de toelichting

- Alle tabellen en bijbehorende kolommen in het PDM zijn toegelicht;
- Grote afwijkingen in het PDM ten opzichte van het CDM worden toegelicht in de ontwerpkeuzes
- Alle ontwerpkeuzes die tijdens het ontwerpen van het PDM aan het licht zijn gekomen staan omschreven. Per ontwerpkeuze wordt er ingegaan op het probleem, welke mogelijke oplossingen er voor waren, wat de voor- en nadelen van elke oplossing zijn en waarom er uiteindelijk voor één oplossing is gekozen.
- Alle integrity rules zijn terug te leiden naar de bijbehorende constraint(s) en business rule(s);
- Alle integrity rules hebben een korte (SMART) omschrijving van de implementatie.
- Alle integrity rules krijgen een naam volgens de conventie <TYPE_CONSTRAINT>_<NAAM_CONSTRAINT> en de naam van de tabel waarop de rule op geïmplementeerd word.
- Alle domeinen uit het PDM zijn toegelicht.
- Voor alle tabellen staan de primary keys genoteerd.
- Het TO bevat een uitleg van hoe de staging area van van data wordt voorzien.

Traceability tussen functioneel en technisch ontwerp

De volgende onderdelen moeten terug te vinden zijn in zowel het functioneel als het technisch ontwerp:

- Business rules die worden toegelicht in het functioneel ontwerp moeten in het technisch ontwerp gebruikt worden voor het uitschrijven van constraints, vervolgens moeten deze constraints worden gebruikt voor het uitschrijven van IR's (integrity rules).
- Concepten uit het CDM moeten terugkomen in het PDM, kolommen worden toegelicht in het FO en de domeinen in het PDM.

Testplan

Het testplan moet voldoen aan een aantal punten:

- Er is omschreven welke software benodigd is voor het testen
- Voor iedere IR wordt omschreven welke scenario's minimaal uitgewerkt moeten worden
- Er is wordt een ruwe schatting gemaakt van de tijd die benodigd is voor het implementeren van de tests

Testrapport

Het testrapport moet voldoen aan een aantal punten:

- De uitslagen van alle unit tests staan gedocumenteerd
- Er is een stappenplan waarin staat hoe je de tests uitvoert

Overig

Lokale functienamen, domeinfuncties en overige functies zijn zo kort en beschrijvend mogelijk.

Bronnen

- *GitLab Team*. (n.d.). SQL Style Guide. GitLab. Retrieved May 9, 2022, from <https://about.gitlab.com/handbook/business-technology/data-team/platform/sql-style-guide> Groep A4. (2022, April 21).
- *Football League Manager* - Testplan. Bitbucket. Retrieved May 9, 2022, from https://isebitbucket.aimsites.nl/projects/S22122A4/repos/football-league-manager/browse/docs/test_plan.md