Een korte beschrijving van de aanpak. Hoe ga je te werk.

De migratie:

Wij hebben ten eerste alle eisen aan het nieuwe systeem doorgenomen. Deze hebben we geprioriteerd voor zo ver dat nog nodig was, aangezien het meeste al gedaan was. Hierna hebben we de klassendiagrammen van beide instanties gemaakt. Deze hebben wij naast elkaar gezet om goed te kijken naar de verschillen en overeenkomsten. Hier hebben we één klassendiagram uit gemaakt door de eerdergenoemde overeenkomsten en verschillen te gebruiken. Dit hebben we gedaan met het gebruik van de eisen aan het systeem, maar alleen met de eisen die wij waar konden maken met de bestaande informatie. De eisen die nieuwe klassen in het klassendiagram nodig hebben, zijn er pas bijgekomen nadat we beide oude klassendiagrammen volledig hebben gecombineerd in het nieuwe diagram.

Na het diagram te maken hebben wij gewerkt aan het datadictionary. Dit was niet zo moeilijk, maar vooral veel werk. De verschillende klassen hadden we al, maar hier moesten we de datatypes, of deze nullable zijn, en enige comments nog bij zetten. Om dit te maken hebben we het nieuwe klassendiagram voor de klassen, en de twee oude data dictionaries voor de datatypes gebruikt.

Na het data dictionary was het RIM aan de beurt. Deze hebben we gemaakt door middel van het data dictionary omdat hierin alles staat wat er voor nodig is. Er staan de data types, nullables, primary en foreign keys in. Het RIM samen met de Datadictionary hebben redelijk wat tijd gekost, omdat aanpassingen steeds twee keer moeten gebeuren.

Een korte beschrijving van de wijze waarop je gaat testen of de migratie juist is verlopen.

We kunnen de migratie op een aantal verschillende manieren testen. Bijvoorbeeld door:

* De eisen aan het nieuwe systeem nog een keer doornemen en testen op het systeem door de eisen in praktijk uit te voeren voor zover dit mogelijk is.
* Test data te gebruiken om alle relaties, constraints en triggers te controleren.