

Analyse

De analysefase is een cruciaal onderdeel van elk softwareontwikkelingstraject. Voor de "Open Days Tracking" functionaliteit binnen TravelMate heb ik deze fase grondig doorlopen. In deze sectie leg ik uitgebreid uit hoe ik te werk ben gegaan en wat de belangrijkste uitkomsten van mijn analyse zijn. Mijn doel is om gedetailleerd te laten zien dat ik een complex vraagstuk kan doorgronden en dit kan vertalen naar concrete, realiseerbare eisen, als fundament voor de verdere implementatie.

Hoe de Analyse tot Stand Kwam:

Mijn analyseproces begon met een grondige kennismaking met het bestaande TravelMate-systeem. Ik heb me hierbij niet alleen verdiept in de beschikbare documentatie, maar heb vooral ook veel tijd geïnvesteerd in het voeren van gesprekken met de mensen die dagelijks met het systeem werken, zoals de product owners en engineers.

Al snel werd duidelijk dat er **een belangrijk onderdeel ontbrak in het huidige systeem**: TravelMate had geen manier om bij te houden of die speciaal toegewezen "Open Dagen" eigenlijk wel gebruikt werden. Dit zorgde voor **verschillende grote problemen** voor TravelMate:

- **Gebrek aan overzicht:** Administrators en reviewers konden niet zien of de 'Open Dagen' gebruikt werden, wat hun vermogen om reispatronen en de daadwerkelijke benutting te beoordelen belemmerde.
- **Inefficiëntie in de middelenallocatie:** Ongebruikte reisdagen bleven onopgemerkt en konden niet efficiënt opnieuw worden toegewezen, wat mogelijk leidde tot verspilling van middelen.
- **Beperkte data-gedreven besluitvorming:** Zonder concrete gebruiksdata was het onmogelijk om de toewijzing van reisdagen te optimaliseren op basis van daadwerkelijke behoeften.

Deze situatie resulteerde in een duidelijke verspilling van middelen en verhoogde compliance-risico's, wat de noodzaak voor een oplossing onderstreepte.

Toen het probleem helder was, heb ik dit concreet geformuleerd in mijn SRS: "Het TravelMate-systeem mist inzicht in het gebruik van 'Open Dagen', wat resulteert in inefficiënte middelenallocatie en een gebrekkige controle op het beleid". Vanuit deze heldere probleemstelling vloeide mijn hoofddoel voor dit project voort: het ontwikkelen van een robuust en gebruiksvriendelijk systeem dat deze 'Open Dagen' nauwkeurig registreert en deze informatie in een overzichtelijk administratief paneel presenteert. Dit paneel moest ook functionaliteit bieden voor het heralloceren van ongebruikte dagen.

Voordat ik begon met het opstellen van de initiële SRS, heb ik een meeting gehad met de product owner om de eerste conceptuele eisen en de scope van het project te bespreken. Tijdens dit overleg zijn de behoeften en functionaliteiten vastgesteld, en is de project scope verder gedefinieerd. Dit overleg en de gemaakte afspraken zijn vastgelegd in mijn [Refinement notes product owner](#), die als essentiële input dienden voor mijn verdere analyse en de eerste versie van de SRS.

Vervolgens ben ik echt aan de slag gegaan met het systematisch verzamelen en specificeren van de eisen, gebaseerd op deze initiële besprekingen. Ik heb veel gepraat met verschillende stakeholders – van de tribe reviewers/admins tot de product owners en zelfs de medewerkers die deze 'Open Dagen'

gebruiken. Hun directe input was essentieel om de uiteenlopende behoeften en verwachtingen grondig in kaart te brengen. De eerste conceptversie van mijn bevindingen, inclusief de eerste set aan requirements, heb ik vastgelegd in de **Software Requirements Specification (SRS)** [SRS OPEN days.docx](#).

Later in het analyseproces, na de totstandkoming van de initiële SRS en verder overleg over de technische implementatie en beschikbare systemen, is een belangrijke wijziging in de bron voor gebruikersdata besloten. In de eerste SRS werd mogelijk nog uitgegaan van AnyOrg, maar na een specifiek overleg met de product owner is besloten om over te stappen op **WFIDB**. Deze beslissing werd genomen omdat AnyOrg primair is opgezet voor het verkrijgen van hiërarchische organisatiestructuren, en hoewel het in TravelMate soms ook voor gebruikersdata wordt ingezet, is WFIDB de daarvoor **bestemde en betrouwbare bron** binnen Swisscom voor medewerkersdata. Dit cruciale overleg en de definitieve beslissing zijn vastgelegd in een screenshot van het chatbericht met mijn product owner: [Screenshot chat product owner.png](#).

Deze iteratieve aanpak, beginnend met de refinement notes, doorlopend in de SRS-ontwikkeling, en inclusief de latere beslissing over de databron, laat duidelijk zien dat ik opensta voor feedback, kritisch kan reflecteren op mijn werk en dit kan verbeteren op basis van ontvangen input en nieuwe inzichten. De wijziging van AnyOrg naar WFIDB is vervolgens consistent doorgevoerd en gedetailleerd beschreven in de definitieve SRS. De uiteindelijke versie van mijn analyse is te vinden als [SRS OPEN days after feedback.docx](#).

Mijn Analysevaardigheden in de Praktijk:

In de [SRS OPEN days after feedback.docx](#) laat ik gedetailleerd zien hoe ik een relevant probleem heb geanalyseerd en dit heb vertaald naar gedetailleerde en bruikbare eisen die als basis dienen voor de ontwikkeling.

De aanleiding en de gedefinieerde probleemstelling, die duidelijk maken waarom deze feature nodig was en welke specifieke zakelijke problemen het oplost (zoals gebrek aan inzicht en inefficiënte middelenallocatie), staan uitvoerig beschreven in Sectie 1.1 "Motivation" en 2.1 "Product Overview and Functionalities" van de SRS. Dit bewijst een helder begrip van de bedrijfskundige context en de impact van het probleem op de dagelijkse operatie.

Ik heb de functionele eisen, oftewel *wat* het systeem precies moet doen, nauwkeurig beschreven in Sectie 3.2 "Functional Requirements" van de SRS. Denk hierbij aan specifieke vereisten zoals:

- Het ophalen van een lijst met 'Open Dagen' inclusief hun status.
- Het bepalen van de status van een 'Open Dag' op basis van boekingsgegevens.
- De cruciale integratie met **WFIDB** voor het ophalen van werknemersgegevens. Dit omvat gedetailleerde eisen voor data retrieval, matching en robuuste foutafhandeling, inclusief validatie, logging, en de mogelijkheid voor handmatige correcties door beheerders om de betrouwbaarheid van de gegevens te waarborgen, ondanks de bekende issues met incorrecte of verouderde data van WFIDB.
- Het identificeren van gebruikte en ongebruikte 'Open Dagen' door kruisreferenties met boekingsdata.
- De functionaliteit voor beheerders om ongebruikte dagen te heralloceren via het adminpaneel.

- Het loggen van alle significante systeemacties en API-aanroepen om traceerbaarheid te garanderen.

Daarnaast heb ik ook de essentiële non-functionele eisen, de 'kwaliteitseisen', vastgelegd in Sectie 3.4 "Non-Functional Requirements" van de SRS. Hierin staan concrete en meetbare afspraken over:

- **Prestatie:** Het admin dashboard moet snel laden (95% van de verzoeken binnen 2 seconden). Queries voor de Open Day status moeten binnen 1 seconde voltooien voor standaard data loads (tot 500 'Open Dagen' per unit). API-aanroepen worden asynchroon afgehandeld om de gebruikersinterface niet te blokkeren.
- **Betrouwbaarheid:** Het systeem moet bijna altijd beschikbaar zijn (99.5% uptime) en statusupdates moeten consistent zijn, zonder duplicaten of conflicten. De feature moet tests bevatten om te verzekeren dat er geen onderbrekingen zijn.
- **Beveiliging:** Alleen geautoriseerde beheerders mogen statusupdates doen en data moet veilig (versleuteld) worden opgeslagen volgens Swisscom's interne beleid.
- **Onderhoudbaarheid:** Ik heb eisen gesteld voor een hoge testdekking (80%+ unit tests) en modulaire code, zodat toekomstige aanpassingen eenvoudig zijn. Tests moeten geautomatiseerd zijn en belangrijke functionaliteiten dekken.
- **Schaalbaarheid:** Het systeem moet met gemak veel 'Open Dagen' (minimaal 10.000) kunnen verwerken over verschillende afdelingen heen, en klaar zijn voor horizontale schaling.
- **Compliance:** Voldoen aan interne IT-beleid van Swisscom is een must met betrekking tot datahandling en retentie (Niet onnodige gebruiksdata opslaan).

Om de complexe logica en de interacties visueel te maken en te verduidelijken, heb ik ook een **use case diagram** gemaakt. Dit diagram, dat de interacties en de flow van de functionaliteit laat zien, is te vinden in de **Analysis** map van mijn Graduate Folder en helpt om het geheel beter te begrijpen, zowel voor ontwikkelaars als voor andere stakeholders.

Tenslotte heb ik ook kritisch gekeken naar de beperkingen en aannames van dit project, die de scope en de haalbaarheid mede bepalen. Deze staan gedetailleerd beschreven in Sectie 2.3 "Constraints and Assumptions" van de SRS. Hierin staan zaken als:

- Het moeten werken binnen een **bestaand TravelMate systeem** dat **refactoring** vereist. Dit beperkt de flexibiliteit voor volledig nieuwe architecturen.
- De **actieve ontwikkeling in hetzelfde gebied door meerdere contributors**, wat zorgvuldige coördinatie en aanpassingsvermogen vereist.
- De **beperkte bestaande geautomatiseerde tests**, waardoor een robuust testing framework moet worden opgezet, inclusief unit, integratie en mogelijk end-to-end tests.
- Een specifieke constraint is de afhankelijkheid van **WFIDB**, inclusief de potentiële uitdagingen rondom API-beschikbaarheid, rate limits, respons tijden en datakwaliteit.
- Het expliciet benoemen van aannames, zoals de ondersteuning voor refactoring en de prioriteit van testen, toont aan dat ik rekening heb gehouden met de praktische uitvoerbaarheid en realistische grenzen van het project.

Kortom, mijn grondige en gestructureerde aanpak in de analysefase, zoals gedemonstreerd door de initiële 'Refinement notes product owner' die de input vormden voor de eerste SRS, de iteratieve ontwikkeling van de **SRS** (van [SRS OPEN days.docx](#) naar [SRS OPEN days after feedback.docx](#)), de

latere beslissing over de databron (gedocumenteerd in [Screenshot chat product owner.png](#)), heeft een solide en transparante basis gelegd voor de implementatie.