Continuous Design

Incrementeel en iteratief ontwerpen met Continuous Design.
Door Bart de Best

Context:
Bij een Midden en Klein Bedrijf (MKB) organisatie was er behoefte aan grip op de software-ontwikkeling. De sturing van de DevOps teams die uitbesteed was ontbrak in het geheel. De opdrachten werden via collaboratie tools gegeven en direct gerealiseerd zonder een ontwerp of een product backlog tool als Jira. Dit gaf de business het gevoel van snel schakelen en direct sturen. Na verloop van tijd bleek echter dat veel zaken niet waren opgeleverd, de opleveringen steeds trager werden en hetgeen was opgeleverd niet voldeed aan de verwachting. Tijd dus voor een stukje control in te brengen in het ontwikkelproces, maar dan wel Lean en Mean.

Deze blog sluit aan bij de blog 'Incrementeel en iteratief plannen met Continuous Planning' maar is zodanig geschreven dat de blogs onafhankelijk gelezen kunnen worden.

Uitdaging:
De uitdaging bij deze organisatie was dat de business direct in de lead was zonder enige kennis van zaken op het gebied van regievoering zoals planning, prioriteit-stelling, acceptatie, en dergelijk. Het invoeren van elke vorm van control moest, hoe Lean en Mean dan ook beargumenteerd worden. Tevens was in deze platte organisatie niemand ergens eigenaar van. Dit betekende dat elke besluitvorming een Poolse landdag vereiste met het hele (vrij grote MT).

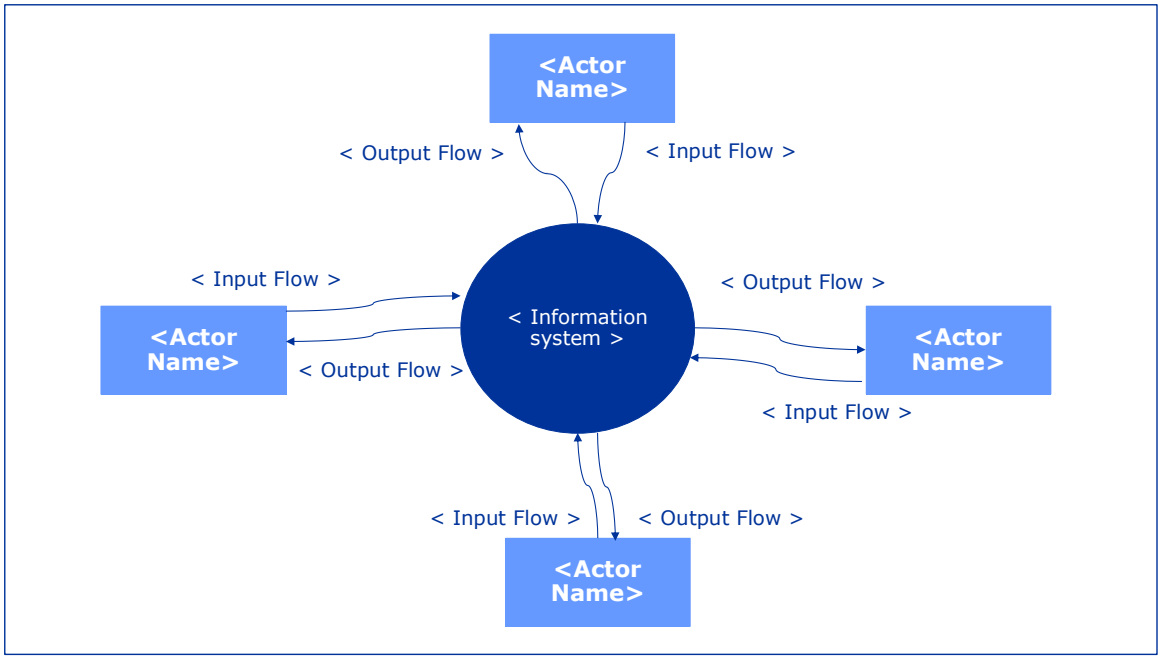
Oplossing:
De oplossing voor deze uitdaging is gevonden in het concept van Continuous Design. Deze blog bespreekt deze aanpak aan de hand van de volgende stappen:

1. Definitie system context diagram
2. In kaart brengen van de value streams
3. Analyse van de value stream middels een value stream canvas model
4. Vertalen van de verbeteringen naar themes en epics
5. Detaillering van het ontwerp in een use case diagram en use cases
6. Completisering door het gebruik van bouwstenen

1. Definitie system context diagram

Als eerste is een overzicht gemaakt van de informatievoorziening middels een system context diagram, zoals weergegeven in figuur 1. In het midden is het informatie-systeem weergegeven.

Een informatiesysteem is hierbij gedefinieerd als een applicatie inclusief de benodigde infrastructurele voorzieningen, plus de value streams die werkzaamheden beschrijven die geautomatiseerd en handmatig plaatsvinden en de gebruiker die dit informatiesysteem gebruiken bij de uitvoering van de werkzaamheden.



Figuur 1, System context diagram template.

De actoren zijn de stakeholders die interacteren met het informatiesysteem. Dit zijn in deze casus het management die financiële doelen definieert en rapportages terugkrijgt. Daarnaast de eindgebruikers die services afnemen. Tevens zijn aanpalende informatiesystemen actoren die informatie met het informatiesysteem uitwisselen. Op deze manier is alle input en output van het informatiesysteem in kaart gebracht. Op zich was dit een prima middel om de scope van het ontwerp snel en goed in kaart te brengen. Het opstellen van dit diagram kostte één uur workshop met drie subject matter experts waarin het diagram is getekend en goedgekeurd. De tekening is in Confluence bewaard en beschreven.

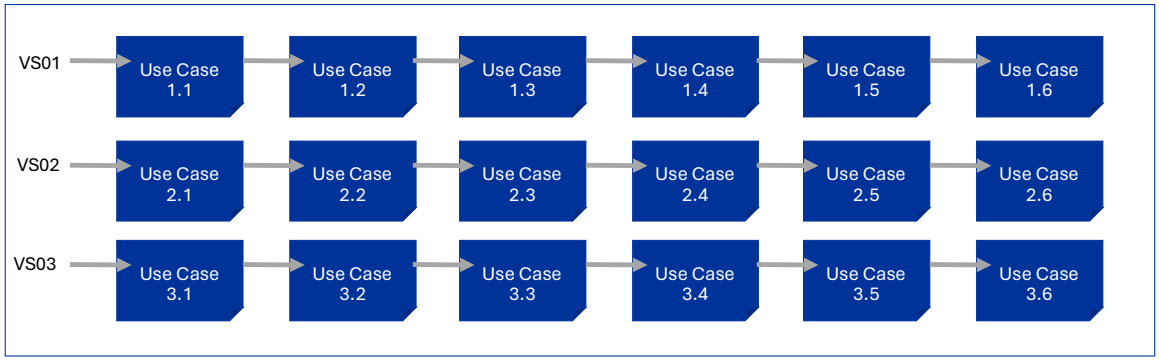
2. In kaart brengen van de value streams

Het system context diagram is gebruikt om de value streams in kaart te brengen die door het informatiesysteem worden ondersteund. Er zijn in totaal 12 value streams onderkend, zoals weergegeven in figuur 2. Een value stream bestaat uit 10 tot 15 stappen die use cases zijn genoemd. De compleetheid van de value streams is gecontroleerd door de input en output van het informatiesysteem van het system context diagram te mappen op de value streams en omgekeerd evenredig. Hierdoor moest het system context diagram worden bijgewerkt omdat er nieuwe input en output stromen naar voren kwamen bij het opstellen van de value streams.

Elke value stream is in een workshop van een uur getekend en gereviewd. Tevens is bepaald wie de eigenaar is van de value stream. Een uitzondering van een belangrijke value stream waarbij de workshop aanpak niet werkte. De betrokken manager kon niet een beeld geven van de werkzaamheden omdat die altijd anders leken te zijn.

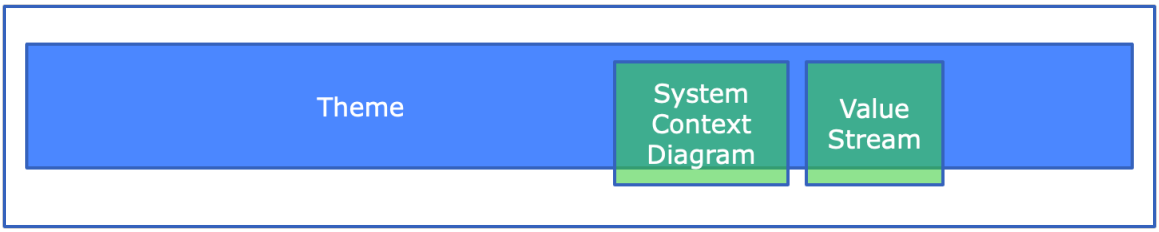
Ook bleek dat er werkzaamheden werden uitgevoerd die niet in het informatiesysteem werden gevangen maar op een whiteboard werden geadministreerd. Na een interview van 4 uur te hebben opgenomen is de analyse van de werkzaamheden

verricht, voltooid en gereviewd. Er was maar één aanpassing nodig en de betrokken manager gaf aan in de afgelopen 5 jaar nog nooit zo'n duidelijk flow van haar werk te hebben gezien. Alle value stream tekeningen zijn in Confluence opgeslagen en beschreven.



Figuur 2, Value stream template.

Op basis van het inzicht dat is verkregen in de administratieve organisatie zijn alle uit staande werkzaamheden gemapt naar de value stream waarvoor ze werden uitgevoerd. Hierdoor kreeg iedere value stream eigenaar een overzicht van wat er speelde. Het MT kreeg een totaal overzicht van alle werkzaamheden. Vervolgens zijn deze werkzaamheden in Jira geadministreerd en gekoppeld aan een thema. Figuur 3 geeft de koppeling van de Continuous Design artefacts (groen) en die van Continuous Planning (blauw) weer.

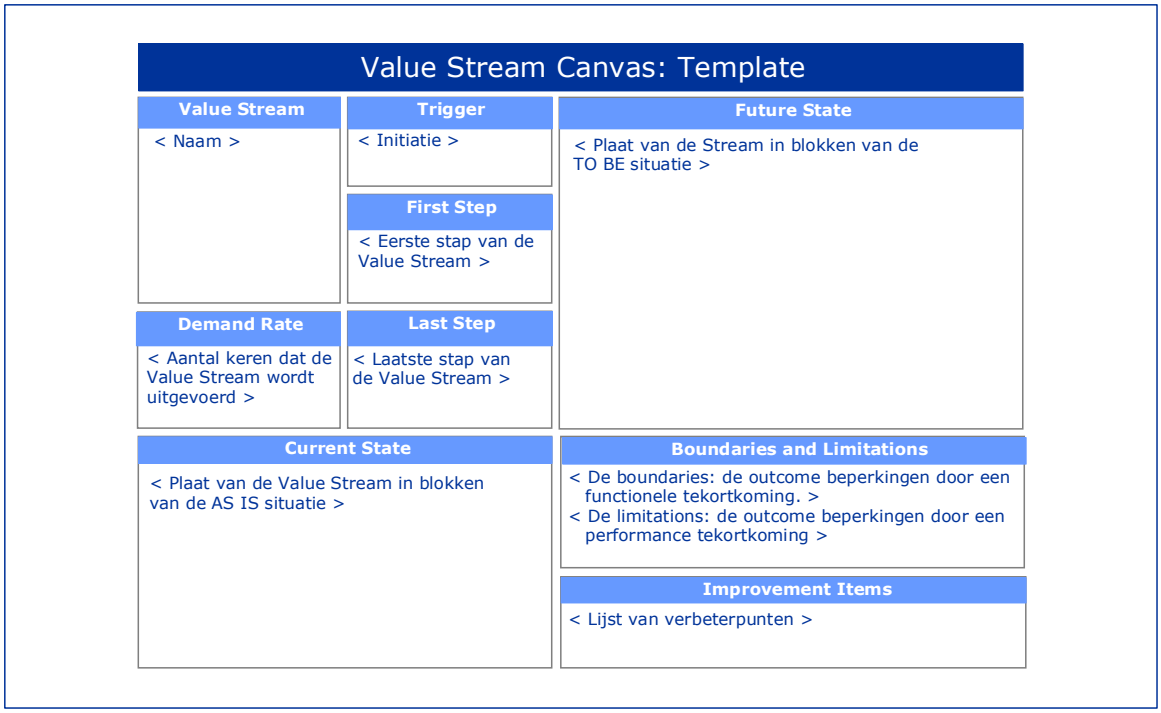


Figuur 3, Continuous design gekoppeld aan Continuous Planning.

Deze rudimentaire besturing gaf erg veel energie, maar was niet genoeg om een overal prioriteit stelling te krijgen.
Parallel aan de stappen 1 en 2 is daarom vanuit Continuous Planning een business analyse verricht naar de missie, visie, bedrijfsdoel en strategie van de organisatie gedaan. Op basis daarvan zijn de themes aangevuld en aangescherpt. Dit is beschreven in de blog "Incrementeel en iteratief plannen met Continuous Planning".

3. Analyse van de value stream middels een value stream canvas model

Na de vaststelling van de themes en value streams is een value stream canvas opgesteld per value stream omdat de belangrijkste bottleneck per value stream te bepalen. Hierbij is een bottleneck gelegen in de performance (limitation) of functionele beperking (boundary). Hiermee zijn verbeterpunten voor de value streams verkregen. De template die gebruikt is per value stream is weergegeven in figuur 4.

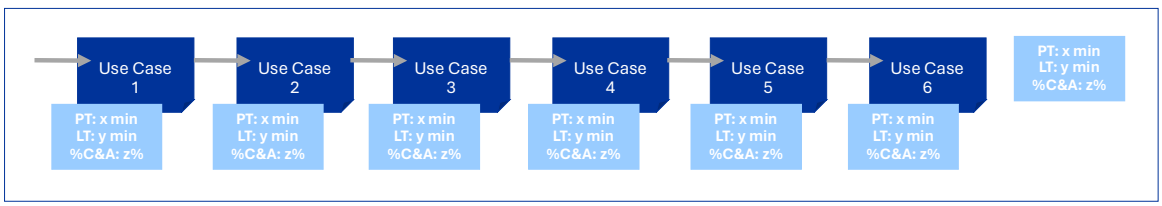


Figuur 4. Value stream canvas template.

Deze template bestaat uit die onderdelen te weten de meta data van de value stream zoals naam, initiatie, eerste en laatste stap. Het tweede onderdeel is de value stream zoals deze is onderkend en de bottlenecks in deze value stream. Het derde en laatste deel is de gewenste situatie nadat de bottlenecks zijn verwijderd.

Elke value stream mapping is in een workshop van een uur tijd verricht en gedocumenteerd in Confluence. De belangrijkste stap was het inschatten van de doorlooptijd (Lead Time = LT), verwerkingstijd (Processing Time = PT) en de kwaliteit (Completeness en Accuracy = %C/A). Dit wordt ook wel Value Stream Mapping genoemd.
De LT, PT en %C/A heten de Lean indicatoren. In eerste instantie werd meestal aangegeven dat dit varieert en dat er geen cijfers bekend zijn. Maar door terug te kijken naar de afgelopen maand en eerst de end-to-end Lean indicatoren in te schatten was dit wel goed mogelijk. De stap met de grootte delta tussen LT en PT werd nader onderzocht op activiteit die dit verschil konden verklaren.

Vaak was dit een wachttijd die verkort kon worden door taken te digitaliseren of de administratieve organisatie aan te passen. Ook de PT tijden die het grootste waren zijn bekeken op verspilling en natuurlijk ook de laagste %C/A. Op deze manier zijn vele issues gevonden in de value streams die mensen op zich wel min or meer wisten maar niet gekwantificeerd op het netvlies hadden en daardoor niet geadresseerd hadden.



Figuur 5. Value stream / Lean indicatoren mapping template.

Een bijzondere situatie was een team waarbij de bottleneck 10% van het aantal FTE kostte. Het betrof het uit e-mail berichten halen van spreadsheets die informatie bevatte die in het informatiesysteem moeten worden ingelezen. Deze bottleneck is er binnen twee weken uitgehaald waardoor de bespaarde 10% van de tijd van dat team besteed kon worden aan andere belangrijke taken waar ze nooit aan toe kwamen. De value stream canvas modellen zijn op Confluence gedocumenteerd en beschreven.

4. Vertalen van de verbeteringen naar themes en epics

De verbeteringen zijn toegevoegd aan Jira in de vorm van theme, epics of features. Een theme is hierbij een verbetering die gelijk staat aan de grootte van je Agile project en een epic is een planningsobject van drie maanden en een feature zes weken. Waar mogelijk zijn de planningsobjecten samengevoegd met de themes uit stap 2.

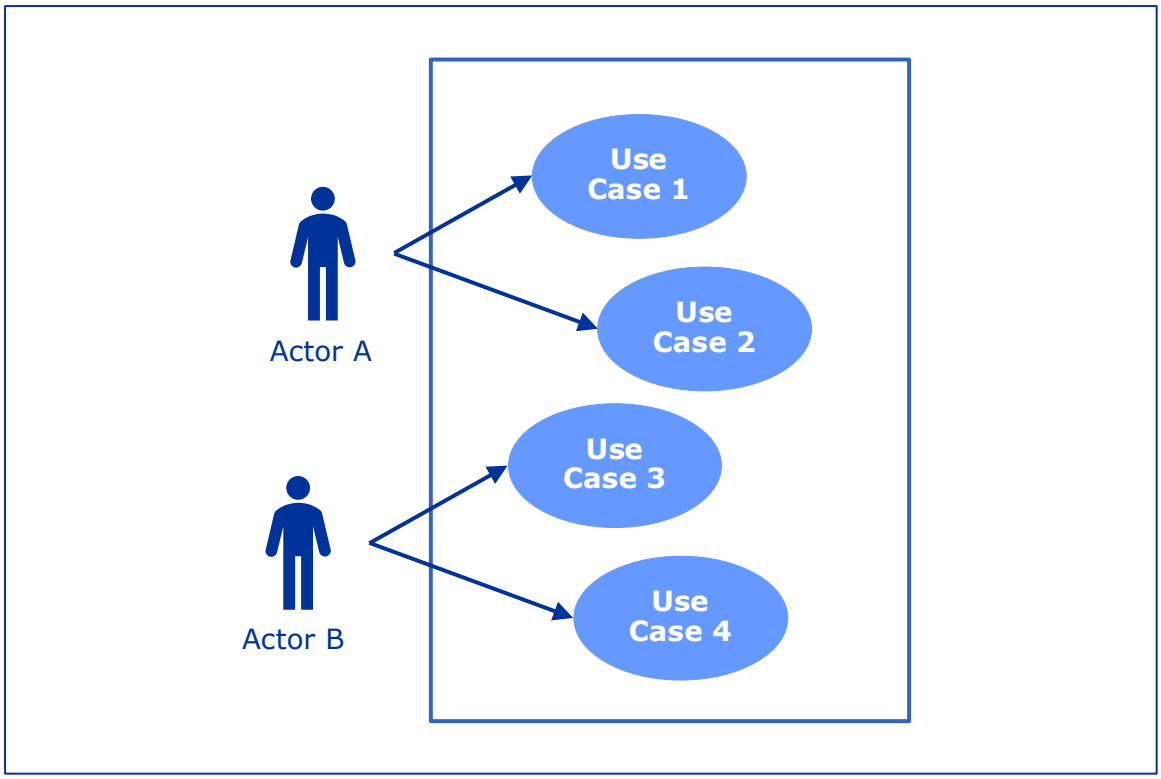
Voor elke epic is een Epic one pager beschreven die de volgende informatie omvat:

- Epic naam
- Epic owner
- Epic doel
- Epic doelgroep
- Epic functies (features)
- Epic hypothese
- Epic risico's
- Epic tegenmaatregelen

Lees ook de blog "Incrementeel en iteratief plannen met Continuous Planning". De uitstaande werkzaamheden (zie stap 2) die al waren gemapt naar value streams zijn of toegevoegd aan de nieuw gedefinieerde planningsobjecten voor de verbete-ringen of zijn voorzien van nieuwe epics en features. Op die wijze is een totale product backlog ontstaan. Kleinere planningsobjecten dan een feature zijn opgenomen als story.

5. Detaillering van het ontwerp in een use case diagram en use cases

De value streams zijn gedetailleerd door een grafische weergaven waarin de actoren en de informatieobjecten aan de value stream stappen zijn toegevoegd. Dit was net iets beter interpreteerbaar voor de business omdat de rollen hierbij tot uitdrukking komen.



Figuur 6. Value stream diagram template.

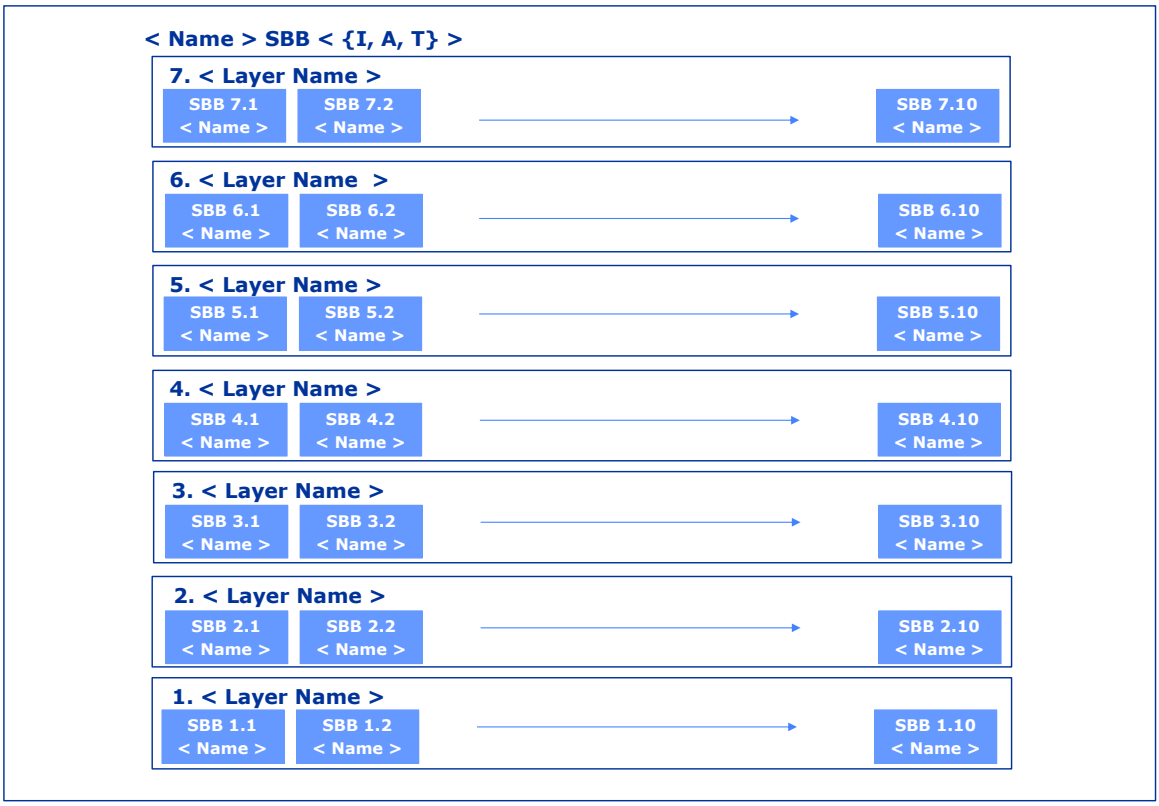
Elke use case is beschreven in een standard formaat. De volgende gegeven worden hierbij gedocumenteerd:

- ID
- Naam
- Doel
- Samenvatting
- Preconditie
- Postconditie
- Performance
- Frequentie
- Actoren
- Trigger
- Scenario met stappen per actor
- Alternatieve scenario's
- Parent use case
- Interface
- Relatie zoals bouwsteen ID

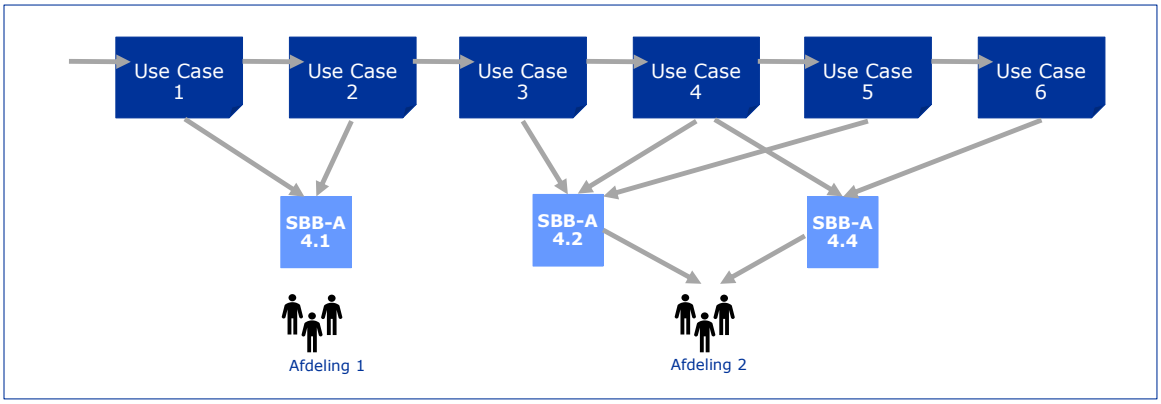
6. Completisering door het gebruik van bouwstenen

Vervolgens is er een applicatie bouwstenenplaat opgesteld waarin de bouwstenen van de applicatie zijn beschreven. Deze plaat bestaat uit lagen die voor elke applicatie generiek zijn. De bouwstenen zijn zelf zijn uniek.

De template is weergegeven in figuur 7. Vervolgens is bepaald welke value stream stap gebruik maakt van welke applicatiebouwsteen zoals weergegeven in figuur 8.



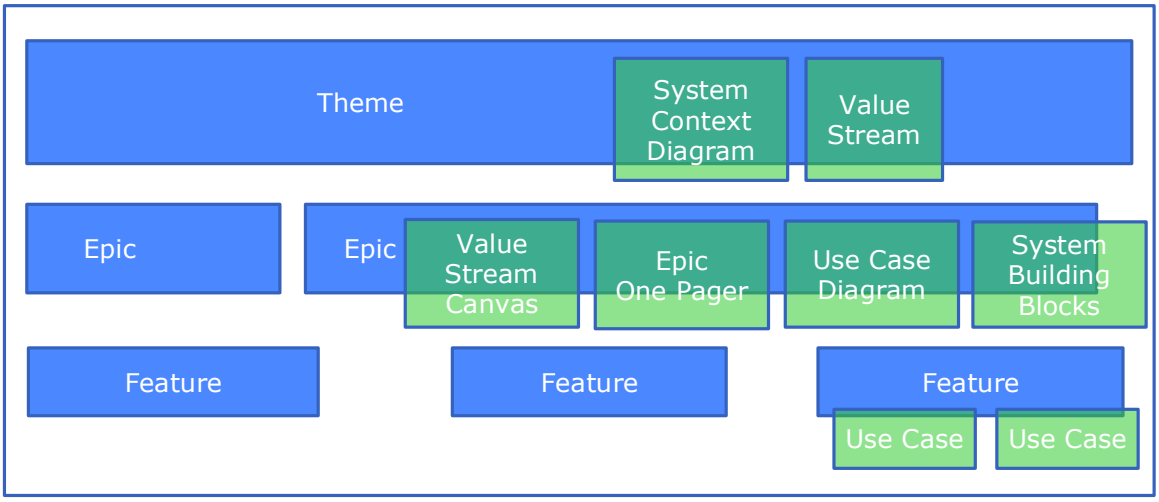
Figuur 7. Applicatiebouwstenen template.



Figuur 8. Value stream / applicatie mapping template.

Op basis van de applicatiebouwstenen plaat kon de compleetheid van de value streams gecontroleerd worden als ook de bouwstenenplaat zelf.

Met het in kaart brengen van de bottlenecks aan de hand van de value stream canvas analyse, de epic one pager, het use case diagram, de use cases en de bouwstenen plaat is de samenhang van Continuous Design en Continuous Planning verder verduidelijkt zoals in figuur 9 is weergegeven.



Figuur 9. Samenhang van Continuous Design en Continuous Planning.

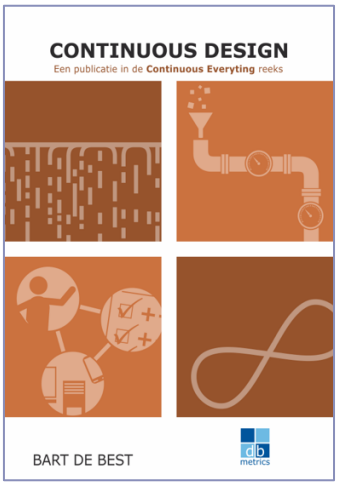
Omdat de planningsobjecten en design objecten consequent zijn gemapt is er een inzicht staan in welke vernieuwing en onderhoud er gaande zijn. Hierdoor kon de organisatie sturing geven door prioriteiten te stellen. Mede door met Continuous Planning het bedrijfsdoel en de strategie te relateren aan de themes en epics ontstond het gevoel van in control zijn.

Doordat de product backlog verfijnd wordt per sprint van de DevOps teams wordt tevens het design verfijnd en groeit dus mee met de planningsdetaillering.
Daarom is dit een goed voorbeeld van de emerging design vormgegeven door de toepassing van Continuous Design.

Bart de Best
G: +31 6 26 55 40 90
E: bart@dutchnordic.group

Leer alles over Continuous Design in de gelijknamige avondtraining. Deze training wordt door Bart de Best gegeven. De eerst volgende avondtraining is op 24, 25, 26 en 27 Juni van 18:30 – 21:30.

Het programma vindt u op:
https://www.dbmetrics.nl/ce-nl/masterclass-continuous-design-nl/



https://www.dbmetrics.nl/ce-nl/continuous-design-nl/