* ~~Context:~~
  + ~~Voor wie?~~
    - ~~Iedereen die het document leest.~~
  + ~~Waarom?~~
    - ~~Zodat de lezer meer inzicht krijgt waar dit document over gaat.~~
  + ~~In welke vorm~~
    - ~~Tekst~~
  + ~~Wat is het product dat gebouwd wordt?~~
  + ~~Hoe past dit product in het huidige systeem????~~
  + ~~Wie gaat de software gebruiken?~~

~~Verplicht~~

* ~~Functional overview~~
  + ~~Voor wie?~~
    - ~~Gebruikers en programmeurs, iedereen die het product zal gaan gebruiken.~~
  + ~~Waarom?~~
    - ~~Zodat de doelgroep inzicht krijgt in de functionaliteiten van de applicatie.~~
  + ~~In welke vorm~~
    - ~~Flowchart met tekst~~

~~Aangeven wat de belangrijkste functies van het product zijn.~~

~~Is het duidelijk wat de user-stories zijn en hoe deze zijn uitgewerkt in het product?~~

~~Is het duidelijk wie de belangrijkste gebruikers van het product zijn?~~

~~Verplicht~~

* ~~Quality attributes~~
  + ~~Voor wie?~~
    - ~~Voor programmeurs en testers.~~
  + ~~Waarom?~~
    - ~~Zo weet je of de functionaliteit aan de kwaliteitseisen voldoet.~~
  + ~~In welke vorm?~~
    - ~~Tabellen~~

~~Quality attributes geven aan wat de kwaliteit van het product is en hoe dit getest is.~~

~~Voorbeelden van quality attributes:~~

~~• Performance (e.g. latency and throughput)~~

~~• Scalability (e.g. data and traffic volumes)~~

~~• Availability (e.g. uptime, downtime, scheduled maintenance, 24x7, 99.9%, etc)~~

~~• Security (e.g. authentication, authorisation, data confidentiality, etc)~~

~~Kwaliteitseisen moeten precies zijn, dus niet: het product werk snel, flexibel, soepel etc.~~

~~Verplicht~~

* Constraints
  + Voor wie?
    - Voor de teamleiders
  + Waarom?
    - Zodat de teamleiders weten wat de projectgrenzen zijn.
  + In welke vorm?
    - Tabellen

Aangeven welke constraints er zijn en welke beslissingen al voor je zijn gemaakt op dit gebied.

Verplicht

* Principles
  + Voor wie?
    - Teamleiders en programmeurs
  + Waarom?
    - Zodat structuur van de werkwijze geëvenaard blijft.
  + In welke vorm?
    - Tekst

De regels aangeven die zijn gemaakt voor het product, zowel de gegeven regels als de regels die het team zelf verzonnen heeft.

Voorbeelden: gebruik React, gebruik Redux, liever bestaande componenten gebruiken dan deze volledig zelf ontwikkelen etc.

Verplicht

* Software architecture
  + Voor wie?
    - Programmeurs
  + Waarom?
    - Om een snel overzicht te krijgen van de structuur van de applicatie.
  + In welke vorm?
    - Van alles, plaatje

De ‘big picture’ van het product aangeven, een algemene overview. Dit kan het beste met container- en componentdiagrammen.

Verplicht

* External interfaces
  + Voor wie?
    - Programmeurs
  + Waarom?
    - Om een overzicht te geven voor de programmeurs welke externe componenten, frameworks etc. Er zijn gebruikt.
  + In welke vorm?
    - Tekst

Dit is voor ons product extreem belangrijk, omdat er behoorlijk wat externe dingen gebruikt zijn, denk hierbij aan alle mogelijke frameworks, React, Redux, Webpack, MongoDB, Bootstrap, maar ook alle componenten en npm-modules die we niet zelf ontwikkeld hebben maar gebruikt hebben van het internet. Hoe zit het met stabiliteit en het updaten van deze componenten?

Volgens de regels niet verplicht maar voor ons wel noodzakelijk.

* Code
  + Voor wie?
    - Programmeurs
  + Waarom?
    - Zodat de herbruikbare functionaliteit duidelijk is voor mensen die nieuw zijn voor de applicatie.
  + In welke vorm?
    - Tekst, diagram, code

De technische specificaties en regels van de meest belangrijke en complexe componenten/software. In ons geval zal het vooral gaan over requests naar de server en hoe er gecommuniceerd wordt met de database. Niet verplicht, maar voor ons wel van relevant.

* Data
  + Voor wie?
    - Doen we niet

Dit onderdeel is niet verplicht.

* Infrastructure architecture
  + Voor wie?
    - Programmeurs
  + Waarom?
    - Zodat duidelijk is welke applicaties op welke locatie draaien.
  + In welke vorm?
    - Afbeeldingen

Een overzicht van hoe verschillende dingen binnen de applicatie met elkaar communiceren: applicatie -> server -> database. Meestal gebeurt dit met diagrammen.

Dit onderdeel is verplicht.

* Deployment
  + Zie hierboven
  + Is me niet volledig duidelijk, wel verplicht
* ~~Operation and support~~
  + ~~Voor wie?~~
    - ~~Programmeurs~~
  + ~~Waarom?~~
    - ~~Zodat programmeurs weten wat ze nodig hebben en wat ze moeten doen om aan de slag te kunnen met de applicatie?~~
  + ~~In welke vorm?~~
    - ~~Tekst~~

~~Uitleggen hoe de software onderhouden wordt. Verplicht~~

* ~~Decision log~~
  + ~~Voor wie?~~
    - ~~Programmeurs~~
  + ~~Waarom?~~
    - ~~Zodat de programmeurs kunnen zien welke beslissingen er zijn genomen en waarom, zodat eerdere fouten niet herhaald worden.~~
  + ~~Welke vorm?~~
    - ~~Tabel~~

~~De keuzes die gemaakt zijn uitleggen. Waarom is er voor een bepaalde userinterface gekozen? Waarom zijn bepaalde frameworks of libraries gekozen?~~

~~Verplicht~~

Vragen/onduidelijkheden:

* Hoeveel hoofdstukken moeten verplicht gedaan zijn?
* Hoe zit het met quality attributes?
* Wie zorgt ervoor na oplevering dat de applicatie up-to-date blijft, dus wie onderhoudt de applicatie?