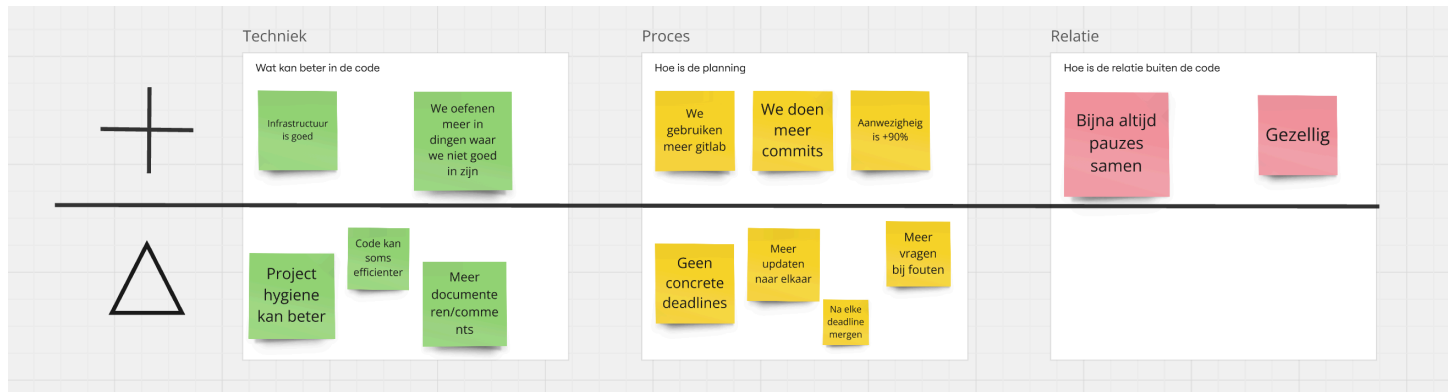


Retro Sprint 4



Actiepunt 1: Duidelijke deadlines maken

Wij willen onze werkwijze verbeteren door duidelijke deadlines te stellen voor onze opdrachten en userstories. Dit helpt ons om overzicht te houden, sneller te werken en problemen op tijd te zien.

Specifiek:

We gaan voor elke userstory en opdracht een deadline opschrijven. Dit doen we samen, zodat iedereen weet hoeveel tijd we hebben om iets af te maken. We zorgen ervoor dat iedereen begrijpt wat er gedaan moet worden en hoeveel dagen het mag duren.

Meetbaar:

Iedere userstory krijgt een aantal dagen waarin het af moet zijn. Als een opdracht klaar is, laten we dit door een teamgenoot controleren en uitleggen. Zo weten we zeker dat het goed is. Als een opdracht niet op tijd af is, bespreken we waarom en kijken we wie uit ons groepje kan helpen om het alsnog af te maken.

Acceptabel:

Het maken van deadlines is haalbaar, omdat we dit in kleine stappen doen. Iedereen in ons groepje begrijpt dat het belangrijk is om afspraken te maken en zich eraan te houden. Als het nodig is, helpen we elkaar om de deadlines te halen.

Realistisch:

Het is mogelijk om voor elke userstory een tijd te geven, omdat we in kleine delen werken. Door samen te kijken naar wat realistisch is, zorgen we ervoor dat niemand te weinig tijd heeft voor grote taken/userstories.

Tijdsgebonden:

Binnen drie weken willen we alles van de sprint af hebben (einde sprint). Vanaf nu gaan we voor iedere nieuwe opdracht of userstory meteen een deadline opstellen en opvolgen. Dit geeft ons meer structuur en zorgt ervoor dat we sneller en meer resultaat zien.

Actiepunt 2: Sneller vragen bij errors

Wij willen sneller om hulp vragen bij errors die we niet binnen één dag kunnen oplossen. Dit helpt ons om minder tijd te verliezen en sneller vooruitgang te boeken in onze opdrachten.

Specifiek:

Wanneer één van ons een error tegenkomt die langer dan één dag duurt, vragen we om hulp binnen ons groepje. Als twee personen het samen niet kunnen oplossen, schakelen we het hele groepje in. Door sneller hulp te vragen, vermijden we dat we vastlopen en kunnen we sneller doorwerken.

Meetbaar:

We gaan bijhouden hoe vaak we hulp vragen en hoe snel errors opgelost worden. Als we merken dat een foutmelding in minder dan twee dagen wordt opgelost door hulp te vragen, weten we dat deze aanpak werkt. Het doel is om errors binnen twee dagen op te lossen en uiteindelijk ook minder errors te krijgen (komt in het volgende actiepunt).

Acceptabel:

Dit is een haalbare aanpak, omdat iedereen in het groepje al bereid is om elkaar te helpen. Samenwerken aan fouten gaat vaak sneller omdat iedereen andere kennis en inzichten heeft.

Realistisch:

Het is realistisch om hulp te vragen bij errors, omdat we al eerder hebben gezien dat samenwerken goed werkt. Door als groep naar de error te kijken, lossen we problemen sneller op en leert iedereen er iets van (Voorbeeld: Rinaldo 1 week bezig met error. Suhayb, Kashif en Maarten 1,5 uur nodig).

Tijdsgebonden:

We willen deze aanpak direct invoeren. Vanaf nu stellen we de regel in dat niemand langer dan één dag met een error worstelt zonder hulp te vragen. Binnen drie weken (einde sprint) kijken we hoe goed deze methode werkt.

Actiepunt 3: Code efficiënter maken

Wij willen onze code efficiënter schrijven door overbodige en onnodig lange stukken code te vermijden. Dit doen we door na elke afgeronde userstory een mini-expert review te houden waarin we samen de code bespreken en verbeteren.

Specifiek:

Na elke afgeronde userstory vraagt degene die de code heeft geschreven aan een teamgenoot om de code te reviewen. De schrijver legt de code stap voor stap uit en bespreekt waarom het op die manier is geschreven. Samen zoeken we naar manieren om de code korter, overzichtelijker en beter te maken.

Meetbaar:

Voor elke userstory houden we een mini-expert review en bespreken we hoe de code verbeterd kan worden. Het doel is om na elke review minstens één verbeterpunt te vinden. Dit zorgt ervoor dat we stap voor stap beter en sneller op een goede manier leren programmeren.

Acceptabel:

Deze aanpak is haalbaar omdat het reviewen van code in kleine stukken gebeurt, direct na een userstory. Iedereen kan zo snel leren van de feedback aangezien het direct invloed heeft op dat stukje code.

Realistisch:

Het is realistisch omdat we in ons groepje al ervaring hebben met elkaar helpen bij code (problemen). Door samen naar de code te kijken, combineren we onze kennis en leren we van elkaars inzichten. Dit zorgt voor betere en snellere code.

Tijdsgebonden:

We gaan deze aanpak meteen gebruiken. Binnen drie weken (einde sprint) willen we dat dit een vast onderdeel is van onze werkwijze. Na elke userstory plannen we ongeveer 10-15 minuten in voor een code-review.