

Met de [verbetervoorstellen](#) in de hand, starten we met de implementatie. De context is nu opnieuw bepalend voor de implementatieaanpak. Omdat grote verbeterplannen in de praktijk weinig succes hebben, kiezen we voor een iteratieve, incrementele aanpak van verbeteren.

### Scrum

Indien de organisatie een kort-cyclisch (voor het gemak even Agile genoemd) ontwikkelproces heeft, bijvoorbeeld met behulp van Scrum, hebben we 'geluk'. De verbetervoorstellen kunnen dan mee in de ontwikkelflow als gepland weer voor het team. De product owner geeft prioriteit en ruimte aan het meenemen van de verbetervoorstellen in de sprintplanning.

### Kanban

Als de organisatie (nog) geen Agile ontwikkelproces volgt, organiseren we de implementatie van de verbetervoorstellen als apart project, maar wel op Agile wijze met bijvoorbeeld Kanban. Hoe dan ook willen we voorkomen dat we teveel tegelijk implementeren of met te lange doorlooptijden, waardoor we het zicht verliezen op de voortgang of waardoor de fut uit het verbeterinitiatief gaat.

We maken de cirkel rond door te kijken naar het [effect van de verbeteringen](#).

---

## Heuristiek | toepassen van verbetering

### Agile verbetering

Projecten die grote wijzigingen in een keer willen implementeren, mislukken vaak. Het toepassen van [agile principes voor software development](#) verhoogt de kans van slagen van de implementatie van verbeteringen. Voorbeeld: *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.* vertaalt zich naar: Verbeteringen worden in regelmatige stappen geïmplementeerd, in een cyclus van weken tot enkele maanden, met een voorkeur voor de kortere tijdschaal. Agile tools zoals kanban-boards en backlogs bewijzen hun nut bij verbeterprojecten.

### Scrum? Dan scrum

Wanneer de organisatie een agile framework zoals scrum toepast voor de reguliere werkzaamheden, dan gaan de verbetervoorstellen als 'user stories' mee op de team backlog. Ze doorlopen dezelfde stappen als reguliere teamtaken, zoals *refinement* en *estimation*.

### Geen scrum? Dan kanban.

Wanneer de organisatie geen goed draaien agile framework heeft voor de reguleren werkzaamheden, dan creëert het implementatieproject een eigen agile aanpak, waarbij kanban goede handvatten biedt.

## Backlog met verbeteringen

Verbetervoorstellen op een backlog worden in volgorde van prioriteit gezet door iemand in de rol van product owner *testverbetering*.

- De verbetervoorstellen doorlopen ook andere agile procedures, zoals 'refinement'.
- Verbetervoorstellen hebben 'acceptatiecriteria' nodig: wanneer is een verbetering succesvol?
- De voorstellen richten zich op het WAT, de teams bepalen het HOE van de uitwerking.

## Langs stapstenen

Het helpt als de implementatieroute opgebouwd is uit stappen waar onderweg al rendement ontstaat op de verbeteringen. Denk hierbij aan stapstenen om een rivier over te steken: ze markeren de route en het motiveert betrokkenen als de sprong naar iedere volgende stapsteen al meteen iets oplevert.

## Product owner testverbetering

De persoon in deze rol heeft de gedelegeerde autoriteit om de verbetervoorstellen een plaats te geven tussen de reguliere taken van de teams die erin betrokken zijn. Hierdoor ontstaat de tijd en ruimte in de teams om de verbeteringen te implementeren. De PO testverbetering beheert de backlog met verbeteringen.

## Verbeteren met de winkel open

Organisaties vinden het moeilijk om verbeteringen te implementeren als dit de productiviteit van het team omlaag brengen. En dat doet het vaak in eerste instantie: de kost gaat voor de baat uit. Betrokken managers hebben een belangrijke taak om in overleg met de (interne) klanten van de teams de condities te scheppen voor een eventuele (tijdelijke) dip in productiviteit. Misschien zijn er ook andere mogelijkheden, zoals het tijdelijk uitbreiden van teams om zo de vaste teamleden ruimte te bieden voor de implementatie van verbeteringen.

## Verbetering-volwassenheid-paradox

Als er veel te verbeteren is, dan zijn er vaak quick wins te behalen en gaat het verbeteren snel, zou men kunnen verwachten. De praktijk werkt andersom. Volwassen organisaties zijn beter in staat verbeteringen te implementeren dan minder volwassen organisaties. Houdt bij minder volwassen organisaties met langere doorlooptijden rekening en plan minder verbetering tegelijkertijd.

## -> naar effect Terugkoppeling (feedback)

De [CDTV plaat](#) laat zien dat CDTV vol zit met feedbacklussen, zoals:

- Lukt de implementatie van de verbetervoorstellen? (Implementatie- Effect - Verbetervoorstellen-Bijgestuurde implementatie).
- Loopt het niet helemaal goed met de implementatie en blijkt dat te komen door een verandering in de context? (Implementatie-Effect-Veranderde context-Aangepaste

aanpak-Hernieuwd onderzoek-Bijgewerkte verbetervoorstellen & implementatie).

Voorbeeld: de organisatie heeft agile werkwijze ingevoerd en de voorstellen gingen uit van een traditionelere werkwijze.

- Het kan zelfs zijn dat de doelstellingen veranderd zijn: bijvoorbeeld de implementatie moet eerder klaar zijn. De hele stack van CDTV-activiteiten moet dan opnieuw worden doorlopen.

## Heuristiek | tot slot

### De Ivoren Toren

Zittend in je Ivoren Toren betrek je de teams die de verbeteringen moeten implementeren vooral *niet* en zendt je directieven uit 'doe het nu maar zo, want ik ben de expert'.

Verbeteringen vanuit een Ivoren Toren hebben in de praktijk een kleine kans van slagen.

### Buy in

Het helpt enorm als gedurende de hele looptijd van het CDTV-initiatief gewerkt is aan het verkrijgen van buy-in voor de voor de verbeteringen noodzakelijke veranderingen. Aspecten die helpen zijn:

- Managers die het belang van de verbeteringen onderstrepen
- Managers die ruimte bieden voor de veranderingen, zoals tijd en budget
- Het is voor betrokkenen duidelijk waarom de verbeteringen nodig zijn.
- Betrokkenen zien voor zichzelf voordelen.

### Geen pijn, geen implementatie

Veranderen gaat normaal gesproken niet vanzelf. Mensen zijn bereid om er tijd, aandacht, inspanning aan te besteden, maar dan moet dat wel iets voor ze opleveren. In de praktijk moet men een bepaalde 'pijn' voelen die weggaat na de implementatie van de verbetering. CDTV-trajecten waaruit blijkt dat er wel verbetervoorstellen zijn, maar eigenlijk geen problemen op te lossen zijn, eindigen met de gezonde conclusie: mooie voorstellen, maar we gaan (nu) niets doen. De voorstellen liggen wel alvast klaar om een pijntje weg te nemen voor als er iets verandert.

### Tips&tricks op een rij

- Knip verbeteringen in stukken die iedereen kan overzien
- Het helpt als ieder deelvoorstel al meteen iets bijdraagt aan de verbeterdoelstellingen of in elk geval voor betrokkenen iets nuttigs oplevert
- Beperkt de hoeveelheid 'verbetering' tegelijkertijd (kanban)
- Stel duidelijke prioriteiten tussen verbeteringen en tussen verbeteren en lopende werkzaamheden
- Evalueer vaak en veel (feedback lussen)
- Goede planning is belangrijk, niet te verwarren met het schrijven van een dik plan

[CDTV plaat.](#)