



Inleiding in projectmethodes Waternal & Agile/Scrum



Graduaatsopleidingen – PXL Digital

2020-2021

Inleiding in projectmethodes waterval & agile/scrum

Om projecten tot een goed einde te brengen, zijn er projectmethodes ontwikkeld. Zo'n methode toepassen, vergroot de kans op het slagen van een project. Het zijn afspraken en richtlijnen over hoe je op een goede en gestructureerde manier projecten kan aanpakken.

Wij behandelen in dit hoofdstuk twee soorten projectmethodes: de klassieke watervalmethode (die bestaat al het langst) en de scrum methode.

Op het einde van deze cursus over projectmethodes, zal je begrijpen wat de typische kenmerken en grote verschillen zijn van waterval t.o.v. scrum, wat scrum betekent in een "agile manier van werken" en wat de belangrijkste principes zijn van scrum.

In een tweede deel (verdieping) komen de rollen in scrum aan bod, geven we meer inzicht in de voor- en nadelen van scrum t.o.v. een klassieke methode, en komen nog enkele bijkomende principes aan bod zoals het inschatten van het werk in scrum.

1.	Klassieke ontwikkeling via de watervalmethode	3
2.	Agile ontwikkelmethode	4
2.1.	Waarom agile?.....	4
2.2.	Wat is Agile? De ervaring.....	5
2.3.	En scrum dan? Wat is scrum?.....	6
	Historiek:	6
2.4.	Vaste gekende termen in scrum.....	7
	BIJLAGE: Woordenschat	10

1. Klassieke ontwikkeling via de watervalmethode



In deze methode voer je een project stap na stap uit in verschillende fases, van analyse tot in productiestelling en onderhoud. Een volgende fase start maar als de vorige fase is afgelopen, goed gedocumenteerd, en liefst ook officieel afgesloten is.

1. **Analyse:** Hier wordt nagedacht en vastgelegd WAT je gaat automatiseren of gaat uitvoeren in het project (bv. wat je gaat maken, installeren, of configureren). Je spreekt met de klant af wat van het project verwacht wordt. De **scope** is het resultaat van een eerste analyse, het is een verzameling van *requirements* (dat zijn eisen) die aangeven wat je wil uitvoeren, implementeren of oplossen in het project. Het beschrijft alles wat de klant wil om het doel (=objectief) van het project te bereiken.
2. **Ontwerp:** In deze fase wordt nagedacht HOE je de oplossing zal maken
3. **Implementatie:** In deze fase wordt het project uitgevoerd, je maakt de oplossing die je tijdens de ontwerpfase bedacht hebt
4. **Testen:** De oplossing moet getest worden. Er wordt gekeken of die voldoet aan de verwachting. Als het goed is en de oplossing aanvaard wordt door de klant, wordt de IT-oplossing in productie genomen of het eindproduct overhandigd aan de klant
5. **Uitrol en onderhoud:** Nadat het product wordt uitgerold (in productie of in gebruik genomen), wordt het eindproduct nog onderhouden, d.w.z. dat fouten of problemen opgelost (kunnen) worden.

De watervalmethode toepassen, betekent dat deze fases van analyse t.e.m. onderhoud fase na fase plaatsvinden, na elkaar, als een “vloeiende waterval”. Een nieuwe fase kan maar starten als de vorige is afgelopen.

Gedurende al die fases wordt aan de oplossing, het eindproduct gewerkt. Dit eindproduct is afgesproken aan het begin van het project.

In het begin van een project wordt er immers afgesproken wat je wil bereiken, m.a.w. het doel (of objectief) van het project.

Voorbeeld : bouw van een huis

Doel/objectief van het project: we willen een nieuwbouwwoning op onze gekochte grond bouwen, een huis waar wij de eigenaar van zijn en waar we met ons gezin minstens 20 jaar kunnen wonen.

Scope: een moderne nieuwbouwwoning met minstens 4 slaapkamers, een bureau, een dubbele garage, een groene tuin.

Timing: we hebben anderhalf jaar, tegen dec 2021 moet het huis instapklaar zijn.

We passen nu op dit project een watervalmethode toe:

Er wordt dus fase na fase gewerkt:

- Analyse: Eerst denken we na over wat we verwachten in het huis en zo definiëren we de *scope*: het huis heeft minstens 4 slaapkamers, een grote woonkamer met open keuken, een bureau voor minstens 2 personen, een muziekkamer waar een vleugelpiano kan staan, een verdieping met minstens 4 slaapkamers en een badkamer.
- Ontwerp: We spreken af met een architect die een tekening/ontwerp maakt van het huis. Hij tekent uit hoe het huis er zal uitzien. Ook een grondplan van de kamers wordt uitgetekend. We keuren het ontwerp goed alvorens er gestart kan worden met de bouw.
- Implementatie: Het huis wordt gebouwd zoals is afgesproken in het ontwerp.
- Test: Met de architect en veiligheidscoördinator wordt er nagekeken of alles veilig en correct verloopt en of alles klopt met de afspraken. Heeft de trap de juiste hellingsgraad? Alles wordt goed nagekeken.
- In gebruik nemen en onderhoud: Als alles klaar is, kan je verhuizen. Er wordt normaal gezien nog een periode van garantie afgesproken met de aannemer of architect.

Dit is duidelijk een watervalmethode. Stap voor stap wordt het huis gebouwd en in orde gebracht. Op een bepaald moment is het volledige einddoel bereikt: het huis is instapklaar.

2. Agile ontwikkelmethode

2.1. Waarom agile?

De watervalmethode heeft als nadeel dat het zou kunnen dat je verwachtingen anders zijn tegen dat je project klaar is, en dat je moeilijk(er) op die veranderingen hebt kunnen inspelen. De projectorganisatie of organisatie zelf zit dan dikwijls stroever in elkaar om met zo'n veranderingen om te gaan.

Agile organisaties spelen gemakkelijker in op veranderingen rondom zich, bij organisaties of op de markt. Veranderingen gebeuren altijd maar sneller en komen meer en meer voor. Als je agile bent, pas je je snel aan aan de wensen van de klant.

Agile werken op projecten betekent niet harder werken maar slimmer werken. Het gaat niet om meer werk te doen in minder tijd maar om meer waarde te creëren met minder werk.

De nadruk ligt op het efficiënt en snel opleveren van producten en/of services die waarde leveren voor de klant.

Als je projecten zou doen waarbij je je baseert op technologie “omdat die nieuw is” of “omdat de medewerkers het tof vinden”, riskeer je dat je eindproduct iets is wat klanten niet willen of waarvoor ze niet willen betalen.

2.2. Wat is Agile? De ervaring

We leggen op een praktische manier uit wat “agile” is aan de hand van een balspel (in de klas) of een Corona-proof variant (online game bv. “game of circles”). Zo ervaar je de principes van agile in de praktijk.

Het balspel:

Dit is een gekende werkvorm die ook bij bedrijven of in opleidingen wordt gebruikt.

We verdelen in teams. Het doel is om als team zoveel mogelijk balletjes door een “proces” te krijgen op 2 minuten tijd.

Je moet daarbij wel de afgesproken spelregels hanteren:

- Elke deelnemer moet elke bal tenminste 1 keer aangeraakt hebben
- De bal moet “air time” hebben bij het wisselen (dus even los zijn van je handen)
- Als de bal op de grond valt, moet hij terug naar de eerste speler
- Je mag een bal niet aan de persoon direct links of rechts van je geven
- Elke bal moet uiteindelijk terugkomen bij de eerste speler

Als team zorg je zelf dat je op een juiste manier het aantal balletjes telt dat je op 2 minuten rond kan krijgen in het team.

Het team doet dit spel een paar keren.

- Het team bepaalt zelf wie de eerste speler is.
- Het team krijgt eerst 2 minuten om je strategie te bespreken.
- Dan start een eerste ronde van 2 minuten ballen doorgeven.
- Het resultaat wordt genoteerd.
- Vervolgens krijg je opnieuw de tijd om met het team de strategie te bespreken om het een volgende ronde beter te doen.
- Deze ronde wordt 3x gespeeld, 3x moet het team zich verbeterd hebben.

Door het spel krijg je inzichten, in wat net met agile te maken heeft:

- **Zelforganisatie** werkt. Je hebt niet altijd een leider nodig die vertelt wat je moet doen. De spelregels waren duidelijk en binnen die spelregels heeft het team zelf bepaald HOE ze het gingen doen.

- Als iedereen zich betrokken voelt en zich engageert, werkt het het best. Zogauw teamleden minder betrokken zijn, is het resultaat een pak minder goed. Iedereen betrokken houden en **vertrouwen** hebben in mekaar dat het goed komt, is belangrijk.
- Lang plannen vooraf heeft niet altijd zin, gewoon DOEN en PROBEREN, opnieuw proberen (in **iteraties**) en jezelf VERBETEREN (=> **Leren en aanpassen**)
- Samen met het team zet je je beste beentje voor om het beste resultaat te halen voor wat gevraagd werd -> **focus op waarde voor de klant.**

2.3. En scrum dan? Wat is scrum?

Agile verwijst naar het “wendbare” van bv. een organisatie of een projectaanpak.

Scrum is een methode om agile te werken, een methodologie die de agile principes volgt.

Als je een scrum werking toepast, is het de bedoeling dat je je aan een aantal principes houdt die scrum voorschrijft. Als je in een “scrum project” terechtkomt, zal je normaal gezien deze principes herkennen (tenzij de organisatie het niet helemaal of anders toepast).

Historiek:

Scrum is een term die afkomstig is uit het rugby. Het is in die sport een manier om een spel te hervatten na een kleine, technische overtreding. Bij een scrum gaan 2 groepen spelers in een voorover gebogen houding tegen elkaar induwen. Het belang van het teamwerk wordt hiermee benadrukt.



De grondleggers van scrum waren Ikujiro Nonaka en Hirotaka Takeuchi. Zij beschreven een productontwikkelraamwerk dat vergeleken wordt met de manier waarop een rugby-team als groep de achterlijn van de tegenstander probeert te bereiken.

Samenwerking, aanpassingsvermogen, snelheid en zelfsturing zijn kenmerken van multidisciplinaire teams die overeenkomen met zo’n rugbyteam.

Begin 1986 werd hierover een artikel gepubliceerd in de Harvard Business Review, dat is een invloedrijk vaktijdschrift over bedrijfseconomie dat publiceerd wordt door de Harvard Business School (in Harvard, nabij Boston).

Naar aanleiding van dit onderzoek ontwikkelde Jeff Sutherland rond 1993 het scrumproces. Ken Schwaber paste dan weer een eigen benadering bij zijn bedrijf toe. Samen werkten ze dit verder uit en in 1995 werd scrum officieel een software ontwikkelmethode.

Scrum is dus ontwikkeld in de USA en het werd eerst vooral gebruikt in Engelstalige projecten, vandaar de Engelstalige terminologie.

2.4. Vaste gekende termen in scrum

Werk je later in een scrumteam in een organisatie, dan zullen normaal gezien de principes toegepast worden die hieronder beschreven staan. Het voordeel van scrum te gebruiken is, dat waar je ook werkt, je altijd dezelfde manier van werken en termen zal horen als het team scrum toepast.

Enkele belangrijke termen zijn:

Sprint: Een sprint heeft een vaste lengte van 1 tot 4 weken (elke organisatie of projectorganisatie kan dat zelf kiezen). In die sprints is het de bedoeling om werkende (software) producten op te leveren. Een sprint is dus een periode waarbinnen je een stukje werkende software of een werkend product maakt.

Voorbeeld van het balspel:

Elke poging om zoveel mogelijk balletjes door te geven is een iteratie of een sprint.

In software ontwikkeling is dit het ontwikkelen van een eerste stukje software.

Sprint review: Een vergadering of afspraak met het team waarbinnen je samen bekijkt welke sprintdoelstellingen gehaald zijn (en welke niet). Dit gebeurt op het einde van elke sprint.

Voorbeeld van het balspel:

Na elke poging werd er gekeken hoeveel balletjes er doorgegeven zijn.

In software ontwikkeling: na een sprint kijken wat er gehaald is.

Sprint retrospective

In een sprint retrospective bespreek je de leerpunten: wat hebben we als team goed gedaan, wat hebben we geleerd, wat moeten we in de toekomst anders doen?

Dit is anders dan een sprint review. In een sprint review kijk je naar WAT je gehaald hebt, in een sprint retrospective blik je terug op HOE je het hebt gehaald en wat je daarbij beter kan doen.

Voorbeeld van het balspel:

Na elke poging besprak het team hoe ze het beter konden. Hoe kon je je manier van werken aanpassen opdat je nog betere resultaten zou behalen?

In software ontwikkeling: leren uit fouten, of hoe iets sneller kan of beter kan en dat bespreken als een stukje klaar is, zodat je ermee rekening kan houden in het vervolg.

Daily standup



Een dagelijkse afspraak met het team waarbij iedereen vertelt

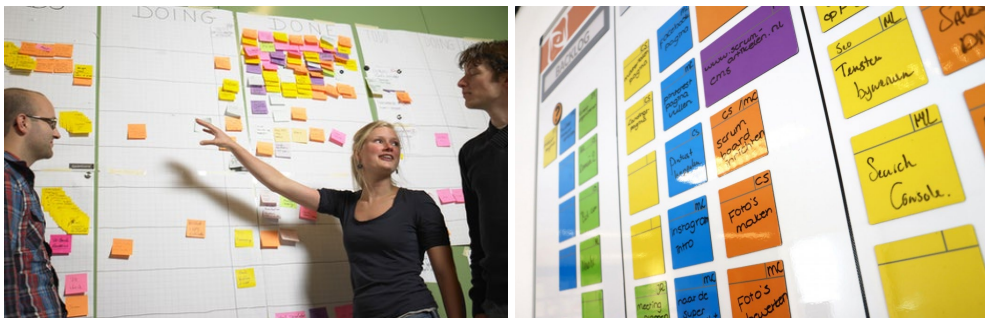
- wat hij/zij heeft gedaan (sinds de vorige daily standup)
- wat je vandaag nog gaat doen (en ev. de volgende dagen)
- wat je uitdagingen zijn hierbij, welke obstakels verwacht worden en hoe het team je daarbij kan helpen

De daily standup wordt georganiseerd door een scrum master. Hij/zij hoeft zelf niet verplicht aanwezig te zijn maar zorgt ervoor dat die daily standups plaatsvinden (en op een efficiënte manier verlopen).

Enkele tips hierbij:

- Je houdt een daily standup kort.
- Begin niet uit te wijden over problemen maar kaart ze aan en maak een afspraak om ze op een ander moment te bespreken met de juiste personen.
- Vertel het aan mekaar, aan alle teamleden, niet aan de scrummaster

Visueel management



Dit is ook heel typisch bij scrum. Taken worden visueel gemaakt, met post-its (op papier of elektronisch). De klant, het management, het team, de product owner en scrum master (rollen waarover je meer leert in de verdieping) hebben allemaal toegang tot een scrumbord met de taken die opgenomen moeten worden (**todo**), waarmee iedereen iedereen bezig is (**ongoing**), en die klaar zijn (**done**).

Visueel management is een krachtige tool bij het toepassen van scrum. Je ziet in één oogopslag hoeveel taken er al klaar zijn of nog lopende zijn. Men heeft ook gemerkt dat het voldoening geeft aan teamleden als ze in een daily standup een taak kunnen verhangen als die klaar is (van “ongoing” naar “done”). Teamleden hebben zo ook een heel visuele manier om aan te duiden waarmee ze

klaar zijn. Je ziet als team dat er resultaat en vooruitgang wordt geboekt, en wat er nog moet gebeuren om alles klaar te krijgen.

Product backlog

De product backlog geeft een overzicht wat je allemaal nog moet realiseren. Het is een lijst van geprioritiseerde punten die moeten ontwikkeld worden om het product te maken en op te leveren.

De punten staan in volgorde van prioriteit. In voorbereiding van een sprint wordt er altijd afgesproken welke punten in een sprint opgenomen zullen worden.

Voorbeeld : consultatie van je lessenroosters

1. Als student wil je je rooster kunnen inzien zodat je weet welke lessen je de komende tijd hebt
2. Als lector wil je je rooster kunnen inzien zodat je weet welk vak je wanneer geeft aan welke klassen.
3. Als student wil je zien van welke lectoren je les hebt.
4. Als student en als lector wil je je lessenrooster kunnen integreren in je Outlook kalender.

Sprint backlog

Dit is de verzameling van taken die moeten gerealiseerd worden in 1 sprint. Dit wordt aan het begin van een sprint afgesproken in een sprintplanningsmeeting.

In het bovenstaande voorbeeld zou het kunnen dat je enkel punten 1 en 2 opneemt in de eerste sprint. Je maakt dan voor die 2 punten gedetailleerde taken die er moeten gebeuren.

Bijvoorbeeld

Bij het eerste punt uit de product backlog:

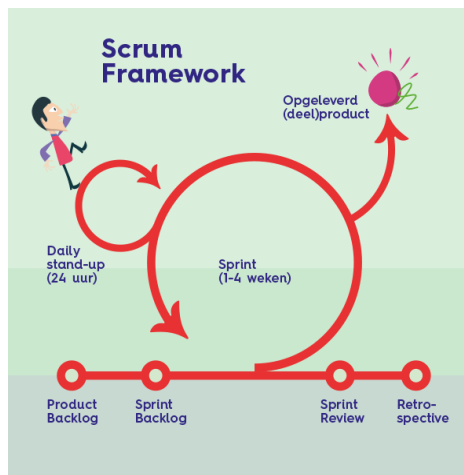
- 1.1. Als student moet je kunnen aanloggen op de applicatie
- 1.2. Het rooster moet getoond worden in een gebruiksvriendelijke kalender
- 1.3. Je kan de kalender bekijken per dag, week of maand.
- 1.4. Je moet kunnen filteren per les, per student of per klas.

Misschien zijn er nog andere taken bv. Als de ontwikkelsoftware nog moet geïnstalleerd worden. Of misschien een infrastructuurtaak dat er nog een development computer moet geïnstalleerd worden met Windows en een programmeersoftware voor de programmeur.

Het tweede punt uit de product backlog kan verdeeld worden in volgende programmeertaken:

- 2.1. Als lector wil je kunnen aanloggen op de applicatie
- 2.2. Als lector wil je het rooster kunnen raadplegen in een gebruiksvriendelijke kalender.
- 2.3. In het rooster staan naam van het vak, klaslokaal en klasgroep vermeld.

Het volgende schema vat de cyclus in scrum samen om een product op te leveren:



Het product dat je wil opleveren wordt in een product backlog beschreven (verschillende “puntjes” (dat noemen we eigenlijk *product backlog items*))

Het (software) product wordt in verschillende iteraties gemaakt. Elke iteratie is een sprint en duurt 1 à 4 weken.

Voor elke sprint neem je enkele punten uit de product backlog die je gaat realiseren in de sprint met het team. Zo maak je een sprint backlog door die punten nog eens uit te splitsen in taken. De taken worden verdeeld onder de teamleden (dat doen ze zelf). Elke dag in een daily standup wordt overlopen hoe ver ze staan (wat is er klaar, waarmee is men nog bezig, waar gaat men aan werken, wat zijn issues om er te geraken).

Op het einde van de sprint houdt het team een sprint review en kijken ze naar wat er gerealiseerd is van de sprint doelstellingen. Ook vindt een retrospective plaats waarin de leerpunten worden besproken.

Na elke sprint kan een deelproduct klaar zijn; na enkele sprints zal het product opgeleverd kunnen worden.

Tot zover de introductie tot scrum (en vergelijking met waterval) en enkele begrippen in scrum.

In een tweede deel zullen we scrum nog verder verdieping door in te gaan op de voor- en nadelen, de rollen in scrum en nog meer.

BIJLAGE: Woordenschat

- **Agile:** agile / agility verwijst naar de “wendbaarheid” van een organisatie, de flexibiliteit om met veranderingen om te gaan.

- **Scrum:** een methode, een framework om op een flexibele manier producten te maken. Er wordt gewerkt in multidisciplinaire teams die in korte sprints werkende producten opleveren. Bij scrum hoort een vaste manier van werken die in scrumteams toegepast wordt.
- **Methodologie:** een verzameling van methodes, een manier van werken die helemaal beschreven staat, procedures en werkwijzes die je kunnen helpen om een activiteit uit te voeren of om kennis te verwerven
- **Multidisciplinair:** verschillende disciplines zijn samengebracht in een team om samen een activiteit te voltooien, dat kunnen verschillende beroepen zijn of verschillende functies/rollen/profielen in een organisatie
 - Een operatie in een ziekenhuis met een multidisciplinair team met daarin een uroloog, een darmspecialist en een gynaecoloog.
 - Een projectteam voor software ontwikkeling met een analist, java ontwikkelaar, database specialist, infrastructuur specialist en 2 testers.