

Sprinten naar succes

Een introductie tot het ontwikkelen
van succesvolle sites en apps



GRATIS
e-book

inspire
digital products
meaningful impact

Inhoud

De rol van de opdrachtgever	4
De product owner	4
Een duidelijk mandaat	4
De beloning	4
Scrum geeft opdrachtgevers de ruimte	5
Ontwikkelen volgens de watervalmethode	5
Ontwikkelen volgens scrum	5
De zes voordelen van scrum	7
1. Scrum levert vaker het gewenste resultaat op	7
2. Scrum heeft een korte time-to-market	7
3. Scrum heeft een hoge return-on-investment	7
4. Scrum maakt voortgang goed zichtbaar	7
5. Scrum leidt tot hoge kwaliteit	8
6. Scrum elimineert overbodige acties	8
De juiste technologie	9
1. Geen dubbele code	9
2. Het wiel hoeft niet opnieuw te worden uitgevonden	9
3. Techniek en design zijn gescheiden	9
4. Snel een werkend prototype	9
5. Goed beveiligd tegen hackers	10
6. Voorbereid op de toekomst	10
Team spirit	11
Interactieontwerper Ruud Vendeloo	11
Visueel ontwerper Heyman Smulders	12
Backend developer David Boot	12
Frontend developer Arco Overbeek	12
Informatieanalist Tim Bakker	13
Tester Michael Veldman	13
Scrum master Anne-Lot Rameau	13
Sprinten naar succes	14
De conceptfase	14
Kick-off	14
Dagelijkse voortgang	15
Sprintdemo	15
Oplevering	15
Winnen op kwaliteit	16
Onderhoudbaar en uitbreidbaar	16
Stabiel	16
Snel	17
Ter zake	17

Succesvolle applicaties ontwikkelen	18
1. Verzeker je van de steun van senior management	18
2. Begin niet te snel met ontwikkelen	18
3. Stel gebruiksvriendelijkheid centraal	18
4. Werk iteratief	18
5. Zet kwaliteit op de agenda	18
6. Luister continu naar gebruikers	19
7. Kies de juiste technologie	19
8. Zorg dat alle kennis gedocumenteerd is	19
9. Zorg voor een realistische planning	19
Tot slot	20
Contactgegevens	20
Woordenlijst	21

De rol van de opdrachtgever

Wie een digitaal product zoals een site of app laat ontwikkelen heeft een verantwoordelijke taak. Niet alleen omdat er een deadline is en een budget, maar vooral omdat iedereen verwacht dat de applicatie een succes wordt. De software moet rekening houden met ieders wensen, snel presteren en er aantrekkelijk uit zien. En mocht dat nodig zijn, dan moet de applicatie later eenvoudig aangepast kunnen worden.

De product owner

Met dit e-book wil Inspire je inzicht bieden in het ontwikkelproces van moderne applicaties. Zodat je maximaal invloed kan uitoefenen. Want je gaat deel uitmaken van ons ontwikkelteam. Dat komt omdat we gebruik maken van een ontwikkelmethode die opdrachtgevers maximaal de ruimte geeft: scrum. Jij beslist. Je wilt een moderne en stabiele app voor pc, tablet en mobiel. Je wilt werken met een ontwikkelteam dat gebruikt maakt van de modernste technologie en alle benodigde expertise in huis heeft. Maar jij hebt de visie. Jij weet wat nodig is. Het is belangrijk dat je jouw kennis en ervaring met ons deelt: wie zijn jouw klanten? Wat willen ze met de toekomstige software bereiken? Tijdens een ontwikkelproject is jouw rol van product owner cruciaal. Jij brengt de wensen van alle stakeholders in kaart. Samen met de informatieanalist vertaal je die naar user stories: beschrijvingen van wat eindgebruikers met de applicatie willen bereiken.

Een duidelijk mandaat

Jouw rol is niet eenvoudig. Die draait om keuzes maken: welke functionaliteit is het belangrijkste en moet als eerste af zijn? Waaruit bestaat de minimale applicatie? Gaandeweg zullen stakeholders met nieuwe ideeën komen. Hoe meer ze van het product in wording zien, hoe enthousiaster ze worden. Hun wensenlijstjes zullen blijven groeien. Het is daarom belangrijk om prioriteiten te stellen. Vaak zul je knopen moeten doorhakken. Daarom heb je een duidelijk mandaat van je organisatie nodig om zelfstandig beslissingen te nemen over de toekomstige vorm en inhoud van de software. Want jouw mening is onze leidraad. Jij bent de spil waar de ontwikkeling om draait.

De beloning

Tijdens het ontwikkelproces zul je voortdurend de vinger aan de pols moeten houden. Zodra het team een onderdeel afgerond heeft, vragen we je om het uit te proberen, eventueel met hulp van eindgebruikers. Is de software daadwerkelijk een uitwerking van de wensen van je organisatie? Je bent veel tijd kwijt aan het informeren van het ontwikkelteam, het beoordelen van versies en het overleggen met je achterban. Al die taken zullen een groot beslag op je leggen. Maar je doet het niet voor niets. Jouw inzet verdient zichzelf terug. De beloning neemt bij elke sprint vastere vormen aan: een app die mooi, stabiel en snel is. Die exact de functionaliteit heeft die nodig is. Die de wensen reflecteert van jouw stakeholders. En die daarom een succes gaat worden.

Scrum geeft opdrachtgevers de ruimte

Traditioneel wordt software in een vaste volgorde ontwikkeld: analyse, ontwerp, implementatie, testen, onderhoud. Je zou denken dat deze zogeheten ‘watervalmethode,’ die in 1970 door W.W. Royce werd geïntroduceerd, opdrachtgevers maximale controle geeft over het ontwikkelproces. Maar in de praktijk blijkt vaak het tegendeel.

Ontwikkelen volgens de watervalmethode

‘In de watervalmethode zijn softwarespecificaties als in beton gegoten,’ zegt oprichter Floris Vlasveld van Inspire. ‘Maar tijdens een webproject ontstaan altijd nieuwe inzichten. Opdrachtgevers denken voor het eerst intensief na over het product dat ze nodig hebben. Ze kijken met een ander oog naar concurrerende applicaties en signaleren marktontwikkelingen waarmee ze rekening willen houden. Ook de gebruikerswensen kunnen om allerlei redenen veranderen. En soms blijken er intern vergelijkbare projecten te lopen waarop de site of app moet aansluiten.’

Voortschrijdend inzicht is normaal. Het heeft niets te maken met slechte voorbereiding, maar alles met een flexibele houding. Maar dan moet een softwareontwikkelaar daar in het ontwerpproces wel rekening mee houden. Traditionele ontwikkelprojecten bieden weinig aanknopingspunten om van mening te veranderen. Vlasveld: ‘Als de opdrachtgever tussentijds een nieuwe versie wenst en het team bouwt software volgens de watervalmethode, dan ontstaat er vaak heftige frictie tussen klant en leverancier. De leverancier kan het ontwikkelproces niet zomaar bijstellen en voelt zich daardoor in het nauw gedreven. Er is pas ruimte voor nieuwe functies als de eerste versie van de software af is. Bij de watervalmethode is dat meestal pas na enkele maanden. Maar dan heeft het project vaak al een verkeerde wending genomen.’

Ontwikkelen volgens scrum

Een alternatieve en sterk aan populariteit winnende ontwikkelmethode is scrum. Deze methode gaat uit van een mondige klant die optimaal meedenkt tijdens het hele ontwerpproces. De benadering werd in 1986 voor het eerst omschreven door Ikujiro Nonaka en Hirotaka Takeuchi. Jeff Sutherland gaf de methode in 1995 de naam scrum. Hij verwees daarmee naar de rugbysport. Daar is een scrum een manier om het spel snel te hervatten.

‘Steeds vaker kiest een opdrachtgever ervoor om eerst met de minimale versie van de site of app de markt op te gaan. Aan de hand van die feedback wordt er dan verder gebouwd.’ — Floris Vlasveld, Oprichter Inspire.

In softwareontwikkeling verwijst scrum naar een iteratieve ontwikkelmethode waarbij geen volledig uitgewerkt ontwerp vanaf de start nodig is. Samen met het ontwikkelteam formuleert de opdrachtgever user stories: beschrijvingen van wat gebruikers met de app kunnen doen. Elke twee weken ontwikkelt, test en presenteert een multidisciplinair ontwikkelteam een nieuwe, werkende versie van de software, op basis van diezelfde user stories. De opdrachtgever ziet zijn site of app dus elke twee weken groeien. En elke twee weken kan hij zijn wensen herijken en aanpassen. Totdat er een versie ligt die precies doet wat nodig is en die klaar is om online te gaan. ‘Steeds vaker kiest een opdrachtgever ervoor

om eerst met de minimale versie van de site of app de markt op te gaan,' zegt Vlasveld. 'Daarna vraagt hij gebruikers wat ze van de software vinden. Aan de hand van die feedback wordt er dan verder gebouwd.'

De zes voordelen van scrum

Scrum leidt tot software van hoge kwaliteit die precies doet wat nodig is. De methode heeft een korte time-to-market. Voortgang is duidelijk zichtbaar en managementkosten blijven beperkt. Hoe kan dat?

1. Scrum levert vaker het gewenste resultaat op

Elk scrumproject start met een brainstorm, waarin het ontwikkelteam achterhaalt wat gebruikers met de applicatie moeten kunnen doen. Die user stories vormen de basis voor de ontwikkeling. Tijdens de daaropvolgende ontwikkelsprints is er ruimte voor voortschrijdend inzicht. ‘Omdat de opdrachtgever de app elke twee weken ziet groeien, komt zij niet voor verrassingen te staan,’ zegt Vlasveld. ‘Mocht een tussentijdse versie afwijken van eerder geformuleerde wensen, dan kan het team bijtijds worden bijgestuurd en de applicatie aangepast.’ Scrum leidt daardoor vaker tot webapplicaties die market fit zijn: ze sluiten aan bij de behoeften van de markt.

2. Scrum heeft een korte time-to-market

‘Met scrum werkt een multidisciplinair team samen in dezelfde ruimte,’ zegt Vlasveld. ‘De feedbackloop is kort, ook tussen opdrachtgever en leverancier. Nieuwe functionaliteit kan meteen worden getest. De opdrachtgever ziet elke veertien dagen een nieuwe versie.’ Scrum kan daardoor tot twee keer sneller zijn dan traditionele ontwikkelmethodes. ‘Die kortere time-to-market ontstaat ook omdat het team met elke sprint opnieuw focust op die user stories die de grootste waarde voor eindgebruikers hebben.’

3. Scrum heeft een hoge return-on-investment

Voorafgaand aan de eerste sprint vraagt het scrumteam aan de opdrachtgever te bepalen welke onderdelen van de software het belangrijkste zijn. Die worden als eerste gebouwd. ‘Nadat uit de brainstorm de user stories naar voren zijn gekomen, schrijven we die op kaartjes. De opdrachtgever moet die kaartjes op volgorde leggen. We vragen de opdrachtgever dus zijn wensen te prioriteren.

‘Met de burn down chart krijgt de opdrachtgever een realistisch inzicht in het moment waarop de applicatie afgerond kan zijn.’ — Arco Overbeek, Frontend developer

Dat dwingt hem om focussen op wat echt belangrijk is.’ Inspire helpt de opdrachtgever eventueel ook bij budgettaire keuzes. ‘De ene user story is bewerkelijker om te ontwikkelen dan de andere. We kunnen voor de klant zichtbaar maken hoe omvangrijk elke wens is door op de kaartjes een t-shirt-maat te schrijven: S, M, L of XL.’

4. Scrum maakt voortgang goed zichtbaar

De product owner van de opdrachtgever is nauw betrokken bij het ontwikkelproces. Hij of zij houdt continu zicht op de voortgang. Onder meer via het scrumboard. Dat laat aan iedereen zien welke user stories nog moeten worden gerealiseerd, welke nu worden gebouwd en welke af zijn. ‘De essentie van Scrum is voor mij de “burn down chart”’, zegt frontend developer Arco Overbeek. ‘Dat is een grafische presentatie die laat zien hoeveel werk in welke tijd wordt verricht. De opdrachtgever krijgt zo een realistisch inzicht in het moment

waarop de applicatie afgerond kan zijn. Bovendien geeft de chart het team een goed idee van de voortgang. Wanneer tijdens de daily standup blijkt dat die onvoldoende is, kunnen de koppen meteen bij elkaar gestoken worden.'

5. Scrum leidt tot hoge kwaliteit

Scrum vereist intensieve samenwerking binnen een team van ontwikkelaars, testers, vormgevers en usability experts. 'Scrum staat of valt met goed overleg,' zegt Vlasveld. 'Het is fantastisch om te zien hoe in een ruimte ontwikkelaars afwisselend gefocust code schrijven – vaak met een goede koptelefoon op – en dan weer samenwerken. Met elkaar, met interactieontwerpers, testers en visueel vormgevers. Je ziet het product daardoor zichtbaar tot bloei komen en in kwaliteit verbeteren.'

6. Scrum elimineert overbodige acties

Doordat teams zelfsturend en multidisciplinair zijn, hoeft er vrijwel geen leiding te worden gegeven. Er bestaat wel een scrum master. Zijn of haar rol is enigszins vergelijkbaar met die van projectleider in traditionele projecten. Maar er is een belangrijk verschil: 'Een scrum master heeft een dienende rol,' zegt Vlasveld. 'Hij moet zorgen voor een prettige werkomgeving en de juiste voorwaarden creëren om aan de slag te gaan. Het team verdeelt zelf de taken.' Overbeek: 'Tijdens de daily standup hoor je met welke werkzaamheden iedereen bezig is. Dat voorkomt dat je software gaat schrijven die iemand anders eerder ontwikkeld heeft. En mocht iemand binnen het team je ergens mee kunnen helpen, dan hoor je dat meteen. Dat scheelt heel erg veel tijd.'

['Omdat de opdrachtgever de webapplicatie elke twee weken ziet groeien, komt hij niet voor verrassingen te staan' — Floris Vlasveld Oprichter Inspire](#)

De juiste technologie

Voor het ontwikkelen van moderne software bestaan verschillende frameworks. Ruby on Rails (RoR) is een framework dat ontwikkelaars helpt om kwalitatief hoogwaardige en beknopte code te schrijven. Het biedt “out of the box” heel veel functionaliteit. Daardoor kunnen ontwikkelaars zich richten op extra's die echte meerwaarde opleveren. RoR vergemakkelijkt het werk van ontwikkelaars en leidt daardoor snel tot een werkend prototype. Hoe kan dat?

1. Geen dubbele code

Ruby on Rails werd in 2004 uitgebracht door David Heinemeier Hanson. Eén van de principes achter dit moderne webapplicatie-framework is Don't repeat yourself (DRY). Dat betekent dat wanneer hetzelfde element op verschillende plekken nodig is in de code, deze niet wordt gekopieerd. In plaats daarvan verwijst de code naar het juiste stukje. Dat voorkomt onnodig werk en fouten en maakt de code eenvoudig onderhoudbaar. 'Ruby on Rails maakt het leven van een ontwikkelaar gemakkelijker omdat dit framework voor webapplicatie-ontwikkeling hem of haar veel taken uit handen neemt,' zegt backend developer David Boot. 'Programmeertaal Ruby is gericht op developer happiness. Het is een heel leesbare taal, bijna Engels voor computers. Dit in tegenstelling tot veel andere programmeertalen: die zijn soms heel abstract en wiskundig. Ruby on Rails dwingt een goede manier van werken af die tot efficiënte code leidt.'

2. Het wiel hoeft niet opnieuw te worden uitgevonden

Een ander principe van Ruby on Rails is Convention over Configuration. Dat betekent dat veel elementen standaard aanwezig zijn. Die bevatten daardoor geen fouten en zijn automatisch goed beveiligd. De ontwikkelaar heeft zijn handen vrij voor de daadwerkelijk uitdagende delen van de code. Boot: 'Dankzij de grote community achter Ruby en Ruby on Rails zijn er veel gems beschikbaar. Dat zijn een soort plugins die een bepaalde functionaliteit zoals bijvoorbeeld authenticatie afhandelen. Die hoeft je alleen nog maar aan te passen aan je eigen wensen. Je hoeft dus niet steeds opnieuw het wiel uit te vinden. Je kunt je daardoor concentreren op functies die echte meerwaarde opleveren voor opdrachtgevers en eindgebruikers.'

3. Techniek en design zijn gescheiden

Ruby on Rails maakt gebruik van een softwarearchitectuur waarbij de voor- en achterkant van de applicatie gescheiden zijn. 'Frontend- en backenders kunnen daardoor aan hun eigen onderdeel werken zonder elkaar in de weg zitten. Tegelijkertijd is het voor beiden eenvoudig om de weg te vinden binnen hun eigen en andermans code, omdat de basisstructuur van de applicatie altijd op dezelfde manier wordt opgebouwd,' zegt frontend developer Arco Overbeek.

4. Snel een werkend prototype

'Omdat je met relatief weinig programmeerwerk toch al snel iets kunt bouwen of wijzigen, is Ruby on Rails een heel flexibel framework,' zegt Boot. 'Ruby on Rails haalt de meest basale taken weg bij de ontwikkelaar. Databasecommando's hoeft je bijvoorbeeld niet zelf te schrijven. Je gebruikt een eenvoudiger equivalent dat Ruby on Rails op de achtergrond

vertaalt in SQL-commando's. Iets soortgelijks geldt voor Javascript. Met die scripting taal kun je bijvoorbeeld animaties en zelfverversende elementen toevoegen aan webpagina's.

Maar Javascript is arbeidsintensief om te schrijven en de code ziet er behoorlijk complex uit. In Ruby on Rails gebruik je Coffeescript. Dat is een vereenvoudigde variant van Javascript. Ruby on Rails zet Coffeescriptcode automatisch om naar Javascript.'

```
square = (x) -> x * x
cube   = (x) -> square(x) * x
```

```
var cub, square;

square = function(x) {
  return x * x;
};
cube = function(x) {
  return x * x;
};
```

Links Coffeescript, rechts gecompileerd Javascript.

'Je hoeft niet steeds opnieuw het wiel uit te vinden. Je kunt je daardoor concentreren op functies die echte meerwaarde opleveren voor opdrachtgevers en eindgebruikers.' — David Boot Backend Developer

5. Goed beveiligd tegen hackers

'Ruby on Rails is een opensource framework,' zegt Boot. 'Iedereen kan de broncode ervan inzien en verbeteringen voorstellen aan de beheerders. Dat gebeurt massaal. Als zoveel ogen meekijken levert dat een enorme versterking en verbetering van de technologie op. Er bestaat een heel grote gemeenschap van ontwikkelaars die zich bezighoudt met het verbeteren van de code. Beveiligingskwesties worden daardoor extreem snel aangepakt.'

6. Voorbereid op de toekomst

Boot: 'Het web is continu in ontwikkeling. Dat moet dus ook gelden voor een webapplicatie-framework. De grootte van de community van Ruby on Rails maakt dat op reguliere basis vernieuwingen worden doorgevoerd. Ruby on Rails is daardoor robuust en op de toekomst voorbereid.'

Team spirit

Scrum is een ontwikkelmethode waarbinnen intensief wordt samengewerkt tussen verschillende disciplines. Een compleet team bestaat minimaal uit de volgende experts: product owner, interactieontwerper, visueel vormgever, backend developer, frontend developer en tester. Soms kan één persoon verschillende rollen op zich nemen, maar vaak niet. Bij complexe projecten kan bovendien behoefte zijn aan extra rollen, zoals een softwarearchitect, kwaliteitscontroleur, systeembeheerder en een databasespecialist.

Product owner Bartho Hengst van KWF Kankerbestrijding:

‘Binnen mijn organisatie ben ik de strategisch manager van KankerNL. Ik ben in 2013 vanaf de start als projectleider betrokken geweest bij de ontwikkeling van kanker.nl. Binnen het scrumteam functioneerde

‘Wat me heel erg aanspreekt van scrum is dat je als product owner continu tussenversies ziet van je product. Het is niet zo dat je over drie maanden wel ziet wat het geworden is.’
— Bartho Hengst Product owner

ik als product owner. Dat betekende dat ik namens KWF Kankerbestrijding onze wensen duidelijk maakte en controleerde of die werden geïmplementeerd. We hadden een proeftuin waarin we eindgebruikers alle mogelijkheden voorleggen. Van welke functies werden ze blij? Die gebruikerservaring vormde de basis voor de ontwikkeling. Ik raadpleegde continu de verschillende stakeholders binnen en buiten mijn organisatie. Maar uiteindelijk bepaalde ik welke user stories het scrumteam tijdens elke sprint uitwerkte. Wat me heel erg aanspreekt van scrum is dat je als product owner continu tussenversies ziet van je product. Het is niet zo dat je over drie maanden wel ziet wat het geworden is.’

Interactieontwerper Ruud Vendeloo:

‘Het interactieontwerp geeft al in een vroeg stadium een goed beeld van hoe de app eruit zal gaan zien. Het beschrijft exact hoe de software zich zal gedragen en hoe een bezoeker de website kan gebruiken. Het is een grafische presentatie van alle onderdelen, zoals knoppen, invoervelden en kaders voor afbeeldingen. Daarnaast toont het van al die elementen welke states ze kunnen aannemen. Voor een winkelmandje is dat bijvoorbeeld leeg of gevuld. Ook laat het interactieontwerp zien welke succesvolle paden een gebruiker binnen de site of app kan doorlopen. Dat is een reeks handelingen die een gebruiker verricht om een taak goed af te ronden.’

‘Om een goede app te bouwen moet je leren denken vanuit de eindgebruiker. Stel een cateringbedrijf wil een website bouwen waarmee potentiële klanten diner-arrangementen voor bedrijfsfeesten kunnen bestellen. Een catering manager zal dan meestal een andere focus hebben dan de typische gebruiker van zo'n website. De manager is gericht op de deal: “Hoeveel gasten komen er? Kan ik een korting op de totaalprijs geven?” Terwijl een directiesecretaresse die de website bezoekt zich eerst zal willen oriënteren: welke soorten voedsel en keukens zijn beschikbaar? Kan er rekening worden gehouden met vegetariërs?’

‘Tijdens de conceptfase werk ik de grote lijnen van het interactie-ontwerp uit. Tijdens de ontwikkelfase vul ik de details in van de user stories die in een nieuwe sprint aan bod komen.’

Visueel ontwerper Heyman Smulders:

'Ik maak het visueel ontwerp. Ik bepaal wat gebruikers uiteindelijk op hun scherm zien: welke lettertypes, kleuren en andere vormgevingselementen. Ik werk op basis van het interactieontwerp. Daarin staat wat er op de pagina moet staan en in welke volgorde. Als het goed is versterkt het visueel ontwerp het interactieontwerp. In mijn ontwerp probeer ik de interactieve elementen naar voren te halen. Zodat meteen duidelijk is waar je op kunt klikken en waarop niet. Dat kan bijvoorbeeld met een kleur of door een knop te maken. Daarnaast probeer ik in mijn ontwerp de focus van de gebruiker te leiden naar de eerstvolgende handeling die hij of zij waarschijnlijk gaat verrichten. Bij een webenquête geef ik bijvoorbeeld de vraag in een opvallend groot lettertype weer. De aandacht gaat daardoor in eerste instantie naar die vraag. Daarna valt het oog pas op de mogelijke meerkeuze-antwoorden. Zo kun je de aandacht van de gebruiker sturen.'

'Ook zonder het visueel ontwerp kan een app technisch goed werken en gemakkelijk te bedienen zijn. Alleen oogt de software dan wat kil. Door de site of app een aantrekkelijk uiterlijk te geven, zal deze de eindgebruiker meer aanspreken. De applicatie zal daardoor intensiever gebruikt worden.'

Backend developer David Boot:

'Frontend en backend developers ontwikkelen samen applicaties. Het werk van een frontend developer spitst zich toe op wat de gebruiker ziet. Hij of zij implementeert het visueel ontwerp. De backend developer zorgt dat de onderliggende gegevens juist worden verwerkt. Wanneer een gebruiker bijvoorbeeld op 'login' klikt, dan controleert de code die de backend ontwikkelaar heeft geschreven of het juiste wachtwoord is ingevuld. Wanneer dat niet zo is, regelt de code van de frontend ontwikkelaar dat er een foutmelding verschijnt.'

'Ik overleg het meest met mijn mede-backend-ontwikkelaars en heb veel contact met de frontend-ontwikkelaars. Die zijn voor de gegevens die online verschijnen van mij afhankelijk. En andersom: frontenders presenteren mijn data op het scherm.'

Frontend developer Arco Overbeek:

'De frontend developer maakt dat deel van de applicatie dat zichtbaar is voor de eindgebruiker. Het is zijn of haar taak om de client side van de applicatie te laten werken zoals bedoeld is. Een applicatie kan nog zo mooi geschreven functies hebben, als die lastig te vinden zijn of niet te gebruiken, dan worden ze alsnog onvoldoende benut.'

'Ik werk veel samen met de visueel vormgever en met de backend developers. Van de visueel vormgever ontvang ik een stijlgids en Photoshop-ontwerpbestanden van alle mogelijk states van de applicatie.'

'Ik probeer de code zo te schrijven dat de software snel reageert. Wanneer er bijvoorbeeld een kleurverloop moet zitten in een webpagina los ik dat het liefst op met een rekenfunctie. Dat werkt meestal sneller dan het laden van een achtergrond-afbeelding.'

'Nog een voorbeeld: wanneer een gebruiker in zijn winkelwagentje het aantal producten terugbrengt naar nul, dan zou het storend zijn wanneer vervolgens de hele webpagina herlaadt. Daarom zorg ik ervoor dat op de achtergrond weliswaar wordt gecommuniceerd met de backend, maar op de voorgrond alleen het aantal bestelde producten wordt ververst.'

Informatieanalist Tim Bakker:

‘Een informatieanalist ondersteunt de product owner inhoudelijk. Al tijdens de conceptfase probeert hij of zij alle details boven water te krijgen die de ontwikkelaars nodig hebben om hun werk te doen. Wat moet het systeem kunnen? Welke formaten hebben de gegevens? Op basis van dat soort informatie schrijft hij of zij de softwarespecificaties. Tijdens elke sprint verfijnt de informatieanalist het functioneel ontwerp verder.’

‘Daarnaast helpt de informatieanalist de product owner slimme afwegingen te maken. Kan een oplossing misschien net iets anders worden geïmplementeerd, waardoor het ontwikkelproces goedkoper wordt? Wanneer slechts drie procent van de gebruikers zit te wachten op een bepaalde functie, dan moet die niet dertig procent van het ontwikkelbudget opslokken. De belangrijkste taak van een informatieanalist is om te focussen op de zakelijke waarde die in elke sprint wordt gecreëerd voor de opdrachtgever. Vanuit dat oogpunt bekijkt hij of zij het ontwikkelproces.’

‘De belangrijkste taak van een Informatieanalist is om te focussen op de zakelijke waarde die in elke sprint wordt gecreëerd voor de opdrachtgever.’ — Tim Bakker, Informatieanalist.

Tester Michael Veldman:

‘Steeds als het ontwikkelteam een nieuwe user story heeft afgerond, controleert de tester of de implementatie voldoet aan de softwarespecificaties die de analist heeft opgesteld. Hij of zij kijkt naar wat er op het scherm gebeurt: voldoen die stappen aan de behoeften van de klant? Als het team ‘testgedreven’ ontwikkelt, is de technische stabiliteit van het systeem al tijdens de ontwikkeling doorlopend gecontroleerd. De tester probeert echter nieuwe manieren te vinden om het systeem onderuit te halen. Een frisse blik en creativiteit zijn daarbij onontbeerlijk.’

Scrum master Anne-Lot Rameau:

‘Ik ondersteun de product owner bij het bewaken van het proces. Ik houd me niet bezig met de inhoud, maar controleer of de ontwikkeling soepel en volgens de scrumprincipes verloopt. Werkt het team stap voor stap de juiste user stories uit? Zijn die gedetailleerd genoeg, zodat de voortgang in de gaten kan worden gehouden? Werkt iedereen optimaal samen? Zijn de wensen van de product owner voldoende duidelijk voor het team? Ik inventariseer continu op de achtergrond of alle benodigde informatie aanwezig is, zodat iedereen zijn werk kan doen.’

Het leuke van scrum is dat je als team echt verantwoordelijk bent voor het eindproduct. Bij elke verzameling user stories die je samen aanpakt geef je een belofte af: deze stories hebben we over twee weken gerealiseerd. Dat kan soms uitdagend zijn, maar je staat er niet alleen voor. Omdat scrum zoveel interactie vereist tussen verschillende disciplines krijg je de kans om van elkaar te leren. Scrum zorgt voor toenadering tussen en een beter begrip van verschillende disciplines. Het zorgt echt voor teambuilding. Een scrumteam bestaat niet enkel uit een verzameling losse eilandjes. Je moet het samen voor elkaar boksen.’

Sprinten naar succes

Scrum-ontwikkeling verloopt in sprints. Dat zijn fases van twee weken waarin het team een nieuwe versie van de site of app ontwikkelt en levert. Maar voordat de eerste sprint van start kan gaan, creëert het team eerst een technisch, visueel en interactieontwerp.

De conceptfase

Inspire start de ontwikkeling van een app altijd met een of meer workshops van een halve tot een hele dag. Tijdens deze conceptfase brainstormen inhoudelijk experts van de opdrachtgever met Inspire over alle functionaliteit die de software moet bevatten. Het concept wordt in detail doorgesproken en de product owner, interactieontwerper en informatieanalist identificeren alle functionele wensen.

‘Tijdens één of meer brainstorm gaan we in eerste instantie op zoek naar epics,’ zegt interactieontwerper Ruud Vendeloo. ‘Een epic is een globale beschrijving van een taak die bezoekers willen volbrengen met de software. Zoals het vinden van informatie over het aanbod, of het plaatsen van een bestelling. Daarna formuleren we per epic de user stories. Dat zijn alle afzonderlijke stappen die nodig zijn om een bepaalde epic helemaal te doorlopen.’

Tijdens de workshop zoekt het team bovendien naar voorbeelden van typische klanten. Vendeloo: ‘We omschrijven deze persona’s zo precies mogelijk. Hou oud is de persoon? Is het een man of een vrouw? Wat kenmerkt de persoon?’ Het zo exact mogelijk typeren van persona’s helpt bij het bedenken van geschikte user stories. Bovendien kunnen we de persona’s later gebruiken om te toetsen of een bepaalde functionaliteit voldoet aan een behoefte. Zit een moeder die online een springkussen voor een kinderfeestje reserveert te wachten op een spaarkaart waarmee ze korting krijgt op volgende bestellingen?’

De epics en user stories vormen de leidraad voor het interactieontwerp. Dat is een grafische weergave van het gedrag van de site of app. Dit interactieontwerp wordt inclusief de uitgewerkte persona’s, epics en user stories tijdens een volgende bijeenkomst aan de opdrachtgever voorgelegd. Na één of meer feedbackrondes heeft die de beschikking over een afgerond ontwerp. Dat vormt een weerslag van zijn wensen, maar biedt wel ruimte voor voortschrijdend inzicht. ‘Binnen scrum is dit oorspronkelijke ontwerp alleen een uitgangspunt. Het team zal het ontwerp van sprint tot sprint verder verdiepen.’

Kick-off

De eigenlijke ontwikkeling start met een zogeheten kick-off, waarin het team wordt samengesteld. De user stories worden nog eenmaal doorgenomen met de product owner. Daarna zet de scrum master ze in de centrale planningssoftware. Elke ontwikkelaar kan vanaf nu op het digitale scrumboard zien welke user stories nog moeten worden gerealiseerd, welke nu worden gebouwd en welke af zijn.

Voorafgaand aan elke sprint vindt er een sprint planning meeting plaats waarin product owner en scrumteam overleggen over de sprint backlog: welke user stories gaan we deze sprint ontwikkelen? ‘We maken eerst een schatting van de complexiteit van de user stories

die we willen uitwerken,' zegt backend developer David Boot. 'Elk van de teamleden kent aan de user story een cijfer toe. Zodra we het eens zijn over de story points kunnen we uitrekenen hoeveel tijd het kost om de nieuwe user story te implementeren. Zo kunnen we de stories verdelen over het team.'

De story points zijn ook een manier om te achterhalen of het project niet aan vaart verliest. 'Ze zijn een maat voor de velocity waarmee het team werkt,' zegt Vendeloo. 'Als die bijvoorbeeld bestaat uit veertig story points per week, dan kun je een realistische schatting maken van de hoeveelheid werk die in de volgende sprint kan worden verzet.'

Dagelijkse voortgang

De dagelijkse voortgang van het project wordt gecontroleerd tijdens de daily standup. Boot: 'Dat is een overleg van vijf tot tien minuten tussen frontenders, backenders, de scrum master en eventueel een tester. Elke aanwezige beantwoordt drie vragen: wat heb ik gisteren gedaan? Wat ga ik vandaag doen? Zijn er zaken die mijn werk in de weg staan? Eventuele problemen kunnen zo op tijd worden gesignaleerd en meteen worden aangepakt. Soms blijkt een user story bijvoorbeeld bij nader inzien voor tweemaal uitleg vatbaar. De scrum master kan er dan in overleg met de product owner en de informatieanalist voor zorgen dat de story net iets duidelijker wordt geformuleerd. Een ander voorbeeld van een obstakel is wanneer de opdrachtgever bepaalde schermteksten nog niet heeft aangeleverd. Ook dan zal de scrum master in overleg treden met de product owner.'

'Een daily stand-up is een dagelijks communicatiemoment waarop je binnen korte tijd heel veel waardevolle informatie ontvangt. Wanneer je hoort dat je collega met een probleem zit dat jij in een ander project al getackeld hebt, kun je bijvoorbeeld voorstellen om even samen achter het scherm te gaan zitten. En andersom.'

Sprintdemo

Na afloop van elke sprint presenteert het team de nieuwe software aan de product owner, zodat deze kan worden geëvalueerd: wat is goed? Wat moet anders? 'Elke twee weken presenteren we een werkende versie van het product aan de klant,' zegt Boot. 'Vaak heeft die aan het begin van het project slechts een vaag idee van wat hij wil. Doordat het team de opdrachtgever blijft bevragen, worden die wensen langzaam exacter. Bij elke sprint krijgt het team feedback. En dat geldt ook voor de klant. Die ziet de applicatie ontstaan en kan ingrijpen wanneer een functie anders dreigt te worden geïmplementeerd dan bedoeld.'

Oplevering

Voordat een nieuwe app live gaat, zet het ontwikkelteam de software in een beschermde acceptatieomgeving. De opdrachtgever kan de applicatie daar testen, bijvoorbeeld met hulp van een aantal eindgebruikers. Wanneer de klant concludeert dat alles klopt, rolt het team de applicatie uit naar de productieomgeving.

Wanneer is een site of app af? Vaak kiest een opdrachtgever ervoor om de webapplicatie zo snel mogelijk live te brengen, op het moment dat deze de minimaal noodzakelijke functionaliteit heeft. Het is op dat moment een minimum viable product. Vendeloo: 'De gebruikersgroep kan er alles mee doen wat nodig is, maar er ontbreken nog toeters en bellen. Het mooie is echter dat zodra de software live staat er vaak waardevolle feedback van gebruikers komt. Zo achterhaal je op efficiënte wijze welke nieuwe user stories

het meest gewenst zijn.’ Ook kan veel worden afgeleid uit de routes die bezoekers door de software volgen. ‘Wanneer websitebezoekers afhaken tijdens een bepaalde stap in een online proces, of slechts een deel van de paden doorlopen, betekent het dat er iets gewijzigd moet worden aan het ontwerp. Dat is het fijne van scrum: die ontwikkelmethode maakt het mogelijk om

‘Een daily stand-up is een dagelijks communicatiemoment waarop je binnen korte tijd heel veel waardevolle informatie ontvangt.’ — David Boot Backend Developer

ook nog na het live gaan van de site of app de prioriteiten te wijzigen en de product backlog te herordenen. Vaak denkt een opdrachtgever dat een bepaalde extra functie heel belangrijk is, maar blijkt er veel meer behoefte aan iets anders te bestaan.’

Winnen op kwaliteit

Een goede site of app is stabiel en snel, gemakkelijk te onderhouden en eenvoudig uit te breiden. Bovendien heeft de applicatie exact die functionaliteit waaraan behoefte bestaat.

Onderhoudbaar en uitbreidbaar

Om allerlei redenen is het belangrijk dat een applicatie gemakkelijk onderhoudbaar en uitbreidbaar is. Tijdens de bouw van software ontstaan vaak nieuwe inzichten. Zodra de applicatie live is, geven eindgebruikers nuttige feedback. Daarnaast kan de markt veranderen, waardoor de opdrachtgever het systeem wil uitbreiden. ‘Goed onderhoudbare code maak je door te kiezen voor een modulaire softwarearchitectuur,’ zegt oprichter Floris Vlasveld van Inspire. ‘Zulke code bestaat uit kleine, afzonderlijke brokken. Dat maakt die code begrijpelijk, minder foutgevoelig en eenvoudig aanpasbaar.’

Tijdens het ontwikkelproces kan een team continu in de gaten houden of nieuwe code aan de eisen van onderhoudbaarheid en uitbreidbaarheid voldoet. Niet alleen door elkaars code regelmatig na te lezen, maar ook met gespecialiseerde code review software. Die analyseert de code en geeft een maat voor de kwaliteit. Wanneer code dubbel voorkomt, een beveiligingsrisico herbergt of een complexe structuur heeft, krijgen de ontwikkelaars daar bericht over. Onvolkomenheden pakken ze daardoor onmiddellijk aan. Dankzij dit soort objectieve methoden ontstaat software die vanaf het allereerste begin voldoet aan de hoogste kwaliteitsnormen.

Stabiel

Ook om de stabiliteit van software te bewerkstelligen bestaan verschillende methoden. Test-Driven Development (TDD) is een ontwikkelmethode voor software waarbij eerst tests worden geschreven en daarna pas de code. ‘Je omschrijft heel exact wat je wilt dat de software doet,’ zegt backend developer David Boot. ‘Steeds nadat je een stukje geschreven hebt, draai je de vooraf gedefinieerde tests.’ De ontwikkelaar kan daardoor geen fouten over het hoofd zien.

Een versiebeheersysteem ondersteunt dit ingenieuze bouwwerk van automatische tests. Het houdt alle wijzigingen in de broncode bij. Zodra de tests aangeven dat één van de ontwikkelaars een fout heeft veroorzaakt, wordt de code teruggezet naar een eerdere versie.

@javascript

Scenario: Sign in with valid email and password

When I visit the login page

And I provide a valid emailaddress and password

And I submit the form

Then I should be signed in

Voorbeeld van een automatische test.

Snel

Op kritieke momenten moet de software goed presteren. Centrale delen van de applicatie kun je daarom onderwerpen aan uitgebreide geautomatiseerde prestatietests. Vlasveld: 'Neem bijvoorbeeld een functie die facturen genereert aan de hand van bestellijsten. Stel je wilt dat het systeem binnen één minuut tienduizend facturen moet kunnen produceren. Die eis kun je op voorhand definiëren in een prestatietest. Tijdens het ontwikkelproces draai je deze test voortdurend, op basis van nepbestellijsten. Zo kun je vanaf het begin waarborgen dat het systeem de prestaties gaat halen die nodig zijn wanneer de applicatie live is.'

Ter zake

'De kwaliteit van een app wordt niet alleen bepaald door de technische maar ook door de functionele eigenschappen,' zegt Vlasveld. 'Om een applicatie te bouwen die voldoet aan de behoefte van eindgebruikers moet je hen betrekken bij elk onderdeel van de ontwikkeling.'

Succesvolle applicaties ontwikkelen

Hoe ontwikkel je een site of app die de verwachtingen overtreft? Negen adviezen van oprichter Floris Vlasveld van Inspire.

1. Verzeker je van de steun van senior management

‘Tijdens de ontwikkeling is de product owner de vertegenwoordiger van de opdrachtgever. Hij of zij is de eigenaar van de uiteindelijke applicatie. Deze persoon moet inhoudelijk deskundig zijn, een goed netwerk hebben binnen alle geledingen van de organisatie, regelmatig contact onderhouden met eindgebruikers en zich verzekerd voelen van de steun van het management. Het belangrijkste is dat de product owner beslissingsbevoegd is, zodat hij of zij knopen kan doorhakken wanneer dat nodig is.’

2. Begin niet te snel met ontwikkelen

‘Sluit de conceptfase niet af voordat het bureau de zakelijke behoefte van jouw organisatie tot in detail begrijpt. Misschien is een extra workshop of deep dive-sessie nodig. Dat levert meer op dan zo snel mogelijk beginnen met de ontwikkeling. De tijd die een verlenging van de conceptfase in eerste instantie extra kost verdien je later met gemak terug.’

3. Stel gebruiksvriendelijkheid centraal

‘Het is lange tijd de trend geweest om zo veel mogelijk functies in een softwarepakket te stoppen. Apps voor smartphones en tablets hebben echter voor een mentaliteitsverandering gezorgd. Eindgebruikers willen tegenwoordig liever een beperkte verzameling functies die fantastisch werkt dan een applicatie waaraan niets ontbreekt. Reserveer om die reden voldoende budget voor het verfijnen van de gebruiksvriendelijkheid. Laat desnoods minder belangrijke functies vallen.’

4. Werk iteratief

‘Kies een bureau dat gebruik maakt van een iteratieve ontwikkelmethode zoals scrum. Het team voegt dan in korte cycli steeds nieuwe functies toe aan de applicatie. Dat leidt tot een korte time-to-market. Scrum is ook van nut wanneer de doorlooptijd van een ontwikkelproject in eerste instantie onduidelijk is. Een complexe marktsituatie kan er bijvoorbeeld voor zorgen dat de invulling van bepaalde vernieuwende onderdelen nog moet worden vastgesteld. Dankzij scrum kan, zelfs wanneer niet alle functionele eisen in kaart zijn gebracht, het project toch van start gaan. Het team focust in elke sprint op die functies die het meeste toevoegen voor de eindgebruiker. Zo plukt de organisatie tijdens elke sprint nieuwe vruchten van het project.’

5. Zet kwaliteit op de agenda

‘Wil je organisatie snel een werkend prototype om een bepaalde aanname te kunnen toetsen? Of zijn de gebruikerswensen grotendeels duidelijk en wil de organisatie een applicatie laten ontwikkelen die voorbereid is op de toekomst? Staar je in dat geval niet blind op een vaste planning, zelfs niet als je interne opdrachtgever haast heeft. Het kan nuttig zijn om voor kritieke onderdelen van de software prestatienormen te formuleren. Het ontwikkelteam neemt die eisen dan mee in de reguliere tests die tijdens het hele ontwikkelproces voortdurend plaatsvinden. Maak duidelijk dat je een applicatie wilt

bouwen die goed onderhoudbaar, uitbreidbaar, snel en stabiel is. En dat je software wilt die ook echt gebruikt gaat worden. Dat laatste bereik je door gebruikers optimaal bij de ontwikkeling te betrekken en rekening te houden met voortschrijdend inzicht. Dat kost in eerste instantie meer tijd, maar die verdien je later terug.'

6. Luister continu naar gebruikers

'Ook voordat de applicatie live gaat kun je eindgebruikers bij het project te betrekken. Bijvoorbeeld door het organiseren van brainstormen waarin focusgroepen vertellen wat zij van de applicatie verwachten. Bruikbaarheidstesten van prototypes helpen om de gebruikerservaring in een vroeg stadium te optimaliseren.'

7. Kies de juiste technologie

'Misschien heeft je organisatie al gekozen voor een bepaalde technologie. Dan kan het desondanks lonen van die keuze af te wijken, zodat de applicatie snel tot wasdom komt en gemakkelijk aanpasbaar is. Een goed ontwikkelteam kent de voor- en nadelen van verschillende ontwikkelmethoden, programmeertalen en platforms. Wanneer nodig kan het team de product owner hierover informeren.'

8. Zorg dat alle kennis gedocumenteerd is

'Omdat er veel verschillende experts aan een applicatie werken, is het belangrijk alle beschikbare kennis te documenteren. Dat garandeert optimale samenwerking tussen het team en goed inzicht in de voortgang van het project. Kennisdeling is daarnaast nodig om de continuïteit van projecten te waarborgen. Mocht een teamlid wegvallen, dan blijft cruciale kennis over de werking van de systemen aanwezig. Ook na de overdracht van het systeem blijft documentatie van waarde.'

9. Zorg voor een realistische planning

'Besef dat het ontwikkelen van gebruiksvriendelijke, betrouwbare software een van de meest complexe taken is die er bestaat. Dat proces vereist de inbreng van een reeks experts. Zij maken in overleg met de product owner het interactieontwerp en het visueel ontwerp. Ze ontwikkelen de serverkant en clientzijde van de applicatie. Ze richten één of meer databases in en testen de applicatie op verschillende browsers en apparaten. Ze tuigen een hostingomgeving op en rollen de applicatie uit binnen de productieomgeving. Al die subtaken moeten met de uiterste zorg worden verricht. Alleen zo kun je op alle fronten een hoge kwaliteit garanderen.'

'Het ontwikkelen van gebruiksvriendelijke, betrouwbare software is een van de meest complexe taken die er bestaat.' — Tim Bakker, Informatieanalist.

Tot slot

We waarderen het bijzonder dat je de moeite hebt genomen deze handleiding voor opdrachtgevers te lezen. We zijn blij dat je met ons wilt werken. Mocht je nog vragen hebben, je weet ons te vinden. Een succesvolle site of app bouwen staat of valt met goede communicatie. Je bent een deel van ons ontwikkelteam.

Veel plezier bij het sprinten naar succes.

Contactgegevens

Inspire Innovation B.V.

Voetiusstraat 2

3512 JM UTRECHT

030 82 00 556

info@inspire.nl

www.inspire.nl

‘Om een applicatie te bouwen die voldoet aan de behoefte van eindgebruikers moet je hen betrekken bij elk onderdeel van de ontwikkeling.’ — Floris Vlasveld Oprichter Inspire

Woordenlijst

backend developer

ontwikkelt dat deel van de applicatie dat onzichtbaar is voor eindgebruikers. Deze backend regelt bijvoorbeeld de communicatie met de database en de verwerking van gegevens.

backlog

verzameling van alle user stories die moeten worden ontwikkeld. Een product backlog beschrijft alle user stories uit het interactieontwerp. Een sprint backlog beschrijft de user stories die het scrumteam tijdens de huidige sprint implementeert.

backlog grooming

vergadering die één of twee maal per sprint plaatsvindt. Hierin worden de story points geschat van user stories uit de product backlog. Zo krijgt de product owner al tijdens de lopende sprint een beeld van de complexiteit van de user stories die in volgende sprints uitgewerkt gaan worden. Omdat er soms tientallen of zelfs honderden ongeschatte user stories zijn, wordt deze meeting meestal beperkt tot een uur.

conceptfase

fase waarin inhoudelijk experts van de opdrachtgever met het bureau brainstormen over de functionaliteit die de webapplicatie moet hebben. Het concept wordt in detail doorgesproken en de interactieontwerper en informatieanalist identificeren alle functionele wensen.

daily standup

overleg van vijf tot tien minuten tussen frontend developers, backend developers, de scrum master en eventueel een tester. Elke aanwezige beantwoordt drie vragen: wat heb ik gisteren gedaan? Wat ga ik vandaag doen? Zijn er zaken die mijn werk in de weg staan? De daily standup zorgt ervoor dat eventuele problemen op tijd worden gesignaleerd en meteen kunnen worden aangepakt.

epic

globale beschrijving van een taak die bezoekers willen volbrengen op de website, zoals het vinden van informatie over het aanbod, of het plaatsen van een bestelling. Een epic wordt altijd opgesplitst in user stories.

frontend developer

maakt dat deel van de applicatie dat zichtbaar is voor de eindgebruiker en waarmee deze de software bedient.

informatieanalist

schrijft de softwarespecificaties: wat moet het systeem kunnen? Welke formaten hebben de gegevens? De informatieanalist ondersteunt de product owner inhoudelijk. Zo helpt hij of zij deze om slimme afwegingen te maken. Kan een oplossing misschien net iets anders worden geïmplementeerd, zodat het ontwikkelproces goedkoper wordt? De belangrijkste taak van een informatieanalist is te focussen op de zakelijke waarde die in elke sprint voor de opdrachtgever wordt gecreëerd. Vanuit dat oogpunt bekijkt hij of zij het ontwikkelproces.

interactieontwerp

beschrijft hoe de (web)applicatie zich zal gedragen en hoe een bezoeker de site of app kan gebruiken. Het is een grafische presentatie van alle onderdelen op het scherm, zoals knoppen, invoervelden en kaders voor afbeeldingen. Daarnaast toont het van al die elementen welke states ze kunnen aannemen. Voor een winkelmandje is dat bijvoorbeeld leeg of gevuld. Ook laat het interactieontwerp zien welke succesvolle paden een gebruiker binnen de applicatie kan doorlopen. Dat is een reeks handelingen die een eindgebruiker verricht om een taak goed af te ronden.

interactieontwerper

maakt het interactieontwerp. Dat is een grafische weergave van het gedrag van de app. Het interactieontwerp wordt inclusief de uitgewerkte persona's, epics en user stories aan de opdrachtgever voorgelegd. Een afgerond interactieontwerp vormt een weerslag van de wensen van de opdrachtgever. De interactieontwerper is niet alleen tijdens de conceptfase, maar ook daarna actief. Het interactieontwerp wordt dan aangepast aan eventuele gewijzigde wensen. Onderdelen die tijdens de conceptfase niet of slechts beperkt aan bod zijn gekomen, worden alsnog in detail uitgewerkt.

kick-off

tijdens de kick-off wordt het team samengesteld. Voorafgaand aan elke sprint is er een sprint planning meeting waarin product owner en scrum team overleggen over de sprint backlog: welke user stories moeten tijdens de volgende sprint worden ontwikkeld?

persona

zo gedetailleerd mogelijke beschrijving van een typische gebruiker van een app. Beantwoordt vragen als: hou oud is de persoon? Is het een man of een vrouw? Wat kenmerkt de persoon? Persona's helpen bij het bedenken van geschikte user stories. Bovendien kunnen ze later gebruikt worden om te toetsen of een bepaalde functionaliteit voldoet aan een behoefte.

product backlog

alle nog niet gerealiseerde user stories.

product owner

vertegenwoordiger van de opdrachtgever en eigenaar van de uiteindelijke applicatie. Deze persoon moet inhoudelijk deskundig zijn, een goed netwerk hebben binnen alle geledingen van de organisatie, regelmatig contact onderhouden met eindgebruikers en zich verzekerd voelen van de steun van het management. De product owner moet beslissingsbevoegd zijn, zodat hij of zij knopen kan doorhakken wanneer dat nodig is. Hij of zij krijgt ondersteuning van de scrum master bij het bewaken van het proces: verloopt de ontwikkeling soepel en volgens de scrumprincipes? Inhoudelijk krijgt de product owner ondersteuning van de informatieanalist: wat moet het systeem kunnen? Welke formaten hebben de gegevens?

scrum

ontwikkelmethode die in 1986 voor het eerst werd omschreven door Ikujiro Nonaka en Hirotaka Takeuchi. Jeff Sutherland gaf de methode in 1995 de naam scrum. Hij verwees daarmee naar de rugbysport. Daar is een scrum een manier om het spel snel te hervatten. In softwareontwikkeling verwijst scrum naar een ontwikkelmethode waarbij geen volledig

uitgewerkt ontwerp vanaf de start nodig is. Samen met het ontwikkelteam formuleert de opdrachtgever user stories: beschrijvingen van wat gebruikers met de webapplicatie kunnen doen. Elke twee weken ontwikkelt, test en presenteert een multidisciplinair ontwikkelteam een nieuwe werkende versie van de site of app, op basis van diezelfde user stories. De opdrachtgever ziet zijn applicatie elke twee weken groeien. Elke twee weken kan hij zijn wensen herijken en aanpassen, totdat er een versie ligt die precies doet wat nodig is en die klaar is om online te gaan.

scrum master

ondersteunt de product owner bij het bewaken van het proces. Controleert of het ontwikkelproces soepel en volgens de scrumprincipes verloopt. Werkt het team stap voor stap de juiste user stories uit? Zijn die gedetailleerd genoeg, zodat de voortgang in de gaten kan worden gehouden? Werkt iedereen optimaal samen? Zijn de wensen van de product owner voldoende duidelijk voor het team? De scrum master inventariseert continu op de achtergrond of alle benodigde informatie aanwezig is, zodat iedereen zijn werk kan doen.

sprint

periode van meestal twee weken waarin een multidisciplinair ontwikkelteam een nieuwe werkende versie van de site of app ontwikkelt, test en presenteert aan de product owner.

sprint 0

de eerste sprint. Start meteen na de kick-off. Tijdens sprint 0 treft het team alle voorbereidingen die nodig zijn: de user stories worden nog eenmaal doorgenomen met de product owner. Daarna zet de scrum master deze in de centrale planningssoftware. Elke ontwikkelaar kan vanaf dat moment op het digitale scrumboard zien welke user stories nog moeten worden gerealiseerd, welke nu worden gebouwd en welke af zijn.

sprint backlog

de user stories die het scrumteam tijdens de huidige sprint implementeert.

sprint demo

presentatie waar het ontwikkelteam het werk van de afgelopen twee weken aan de product owner demonstreert. In feite is het daarmee de afsluiting van de sprint.

sprint planning meeting

vergadering die voorafgaat aan elke volgende sprint en waarin product owner en scrumteam overleggen over de sprint backlog: welke user stories moeten tijdens de volgende sprint worden ontwikkeld?

story points

maat voor de complexiteit van afzonderlijke user stories. Story points helpen de product owner om user stories niet alleen te prioriteren op zakelijke waarde, maar ook op de kosten die nodig zijn om deze te implementeren. Story points helpen daarnaast om de werklast te verdelen en de voortgang van het project te controleren. Het schatten van het aantal story points gebeurt tijdens de backlog grooming of de sprint planning meeting. Elk van de leden kent op voorhand aan elke user story een cijfer toe. Wanneer het team het eens is over de story points kan het, op basis van de velocity, uitrekenen hoeveel tijd het kost om de nieuwe user story te implementeren.

user story

microverhaal dat vertelt wat een bepaald type gebruiker met een onderdeel van de software kan doen. Een user story heeft meestal deze vorm: als gebruiker kan ik iets doen, zodat ik een doel bereik. Voorbeeld: als bezoeker kan ik een datum en een tijd selecteren, zodat ik de beschikbaarheid van de kartbaan kan controleren.

Een user story is in feite een kapstok om het gesprek tussen opdrachtgever en ontwikkelteam te faciliteren. Er wordt vaak aanvullende informatie aan de user story toegevoegd om de details te verduidelijken. De informatieanalist speelt daarin een belangrijke rol.

velocity

aantal story points dat gemiddeld wordt afgerond binnen een bepaalde tijd door het team. De velocity is een manier om te achterhalen of het project niet aan vaart verliest. Het is bovendien een middel om het aantal user stories te voorspellen dat het team in de volgende sprint kan uitwerken.

visueel ontwerp

grafische uitwerking van het interactieontwerp. Datgene wat de eindgebruiker op zijn of haar scherm ziet.