

Er zijn maar weinig organisaties die actief bezig zijn hun hele bedrijf agile in te richten. Lean daarentegen wordt vaak wel in de hele organisatie ingezet. Een legitieme vraag is daarom: Wat kan ik met Lean en hoe is het te combineren met Scrum? Henk Jan Huizer en Rini van Solingen beschrijven hoe Scrum en Lean elkaar versterken.

door: HENK JAN HUIZER & RINI VAN SOLINGEN

Lean maakt niet concreet wat er moet gebeuren, Scrum wel

# LEAN EN SCRUM: EEN SIAMESE TWEELING?

**V**eel organisaties zijn druk bezig om agile (in het bijzonder Scrum) te implementeren. Toch zijn er maar weinig organisaties die hun hele bedrijfsvoering agile inrichten. Door veel directies worden wel vaak organisatiebrede verbeterprogramma's opgestart, vaak met behulp van de 'Lean'-aanpak. De IT-manager is daarom niet te benijden: net bezig om op een agile-manier, iteratief, software te ontwikkelen, dan komt Lean daar ook nog eens bij. Wat is het verschil? Kunnen de twee aanpakken elkaar aanvullen?

Scrum is een raamwerk voor softwareontwikkeling dat via korte iteraties (sprints) van enkele weken steeds weer werkende software oplevert die telkens de hoogste toegevoegde waarde heeft. Scrum is een eenvoudig raamwerk bestaande uit drie rollen (productowner, Scrum-master en teamleden), vier meetings (sprintplanning, daily stand-up, sprintreview en sprintretrospective), en vier artefacten (werkend product, productbacklog, sprintbacklog en definition-of-done). Lean-softwareontwikkeling bestaat uit zeven principes waarmee een organisatie haar proces en producten kan optimaliseren. Lean bestaat vooral uit principes en maakt niet concreet wat er precies wanneer moet gebeuren. Scrum daarentegen doet dat wel. Scrum is veel methodischer dan Lean: het legt een raamwerk neer hoe je als organisatie op een agile-manier software ontwikkelt.

## Risico

Als we Lean en Scrum met elkaar vergelijken, dan valt al snel op dat Scrum gebaseerd is op veel van de Lean-principes. In het kader geven we een overzicht van de zeven principes voor Lean-softwareontwikkeling. Wat daarin vooral opvalt is dat Scrum weliswaar methodisch is en helpt bij het hoe, maar dat het waarom achter de Scrum-elementen niet of nauwelijks expliciet is gemaakt. Als er met Scrum wordt gewerkt dan worden wel veel Lean-principes afgedekt. Helaas is dat dan wel impliciet en voor veel mensen vaak onbewust. Het grote risico is dan dat Scrum mechanistisch wordt toegepast. Men doet wat er staat, maar heeft geen idee waarom. In de praktijk gebeurt

dat regelmatig. Er is zelfs een naam voor: 'Zombie-Scrum', teams die zonder na te denken een set van rituelen plichtmatig uitvoeren. Zonder na te denken waarom ze dat doen en wat ze er eigenlijk mee willen bereiken. Kortom, het antwoord is dus niet Lean of Scrum, maar: beide. De één gaat namelijk perfect over het wat en waarom, terwijl de ander concreet maakt hoe. De projectie van Scrum op de Lean-principes (zie kader), laat precies zien waar de logische combinatie van Lean en Scrum te vinden is. Lean gaat over het waarom achter de onderdelen van Scrum. We mogen dan ook gerust stellen dat Scrum feitelijk een methodische implementatie is van Lean. Veel van de Lean-principes worden in Scrum afgedekt. Echter, het is belangrijk om elk van de zeven Lean-principes expliciet de aandacht te geven die ze verdienen. Een te sterke methodische blik heeft een verhoogd risico op zombiedrag.



Henk Jan Huizer is agile-coach en gecertificeerd Scrum-trainer bij Prowareness (h.huizer@scrum.nl). Rini van Solingen is CTO bij Prowareness (rini@scrum.nl) en deeltijdhoogleraar Global Software Engineering aan de Technische Universiteit Delft (voor meer informatie zie: [www.scrum.nl/site/Whitepapers-agile-scrum](http://www.scrum.nl/site/Whitepapers-agile-scrum)).

HET GROTE RISICO IS DAT SCRUM MECHANISTISCH WORDT TOEGEPAST

## SCRUM IS FEITELIJK EEN METHODISCHE IMPLEMENTATIE VAN LEAN

ag  
Voor reacties en nieuwe bijdragen van IT-experts: Henk Ester, 070-378097

### LEAN-SOFTWARE-ONTWIKKELING



### HOE MAPT LEAN OP SCRUM?

#### 1 ELIMINEER VERSPILLING

Alles wat niet direct bijdraagt tot klantwaarde is verspilling. Voorbeelden hiervan zijn: niet gebruikte functionaliteit, bugs, herhaalde kennisoverdracht, onnodig wachten en dergelijke. De kunst is om verspilling te herkennen. Een belangrijk hulpmiddel hiervoor is het maken van een waardestreammodel.

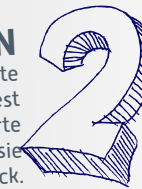


#### ELIMINEER VERSPILLING

- De productbacklog is gesorteerd op waarde en voorkomt dat tijd wordt besteed aan minder belangrijk werk.
- Korte iteraties (sprints) zorgen voor een vast ritme en flow in het proces.
- Zelfsturende crossfunctionele teams voorkomen het doorgeven van werk, wachttijden en onvolledige overdracht van kennis.

#### 2 VERSTERK LEEREFFECTEN

De beste manier om software te ontwikkelen is door het te benaderen als een continu leerproces. Leren lukt het best als er zo snel mogelijk feedback tot stand komt. Korte iteraties waarmee telkens een nieuwe werkende versie wordt geleverd, is een basisvoorwaarde voor feedback. Aannames kunnen dan snel worden getoetst.



#### VERSTERK LEEREFFECTEN

- Elke sprint wordt afgesloten met een review waarin stakeholders feedback geven op het product.
- Elke sprint wordt afgesloten met een retrospective waarin gekeken wordt naar verbeterpunten.
- Elke meeting wordt gedaan met het hele team om te leren en kennis vanuit verschillende disciplines te delen.

#### 3 BESLIS ZO LAAT MOGELIJK

Hoe complexer een project is, hoe beter het is om keuzes uit te stellen en opties open te houden. De beslissing wordt immers beter als er meer informatie beschikbaar is. Iteratief opleveren helpt hierbij. Feedback maakt zaken namelijk duidelijker en helpt om betere keuzes te maken. Zaken die nog onduidelijk zijn kunnen later worden opgepakt wanneer ze duidelijker zijn.



#### BESLIS ZO LAAT MOGELIJK

- Door te werken in sprints wordt precies genoeg tijd gestopt in de voorbereiding van komende sprints en wordt voorkomen dat er tijdro-vende detailbeslissingen over werk in de toekomst worden genomen.
- Elke sprint eindigt met een review van werkende software aan eindgebruikers waardoor voortschrijdend inzicht wordt meegenomen in komende sprints.
- De productbacklog staat niet vast: alles kan er op en er ook weer af. Hierdoor worden beslissingen pas genomen als het echt moet.

#### 4 LEVER ZO SNEL MOGELIJK

Hoe sneller waarde wordt geleverd, hoe sneller men naar de markt kan en hoe eerder er feedback komt om het product nog beter te maken. Daarnaast zorgt sneller leveren ervoor dat het veel makkelijker wordt om beslissingen uit te stellen. Bij langzame processen is het noodzakelijk om ver vooruit te plannen, veel eerder al keuzes te moeten maken, waardoor de flexibiliteit afneemt en het langer duurt om op verrassingen te reageren.



#### LEVER ZO SNEL MOGELIJK

- Korte iteraties (sprints) die telkens een werkende en geteste versie opleveren, vormen het kloppend hart van Scrum.
- Een dagelijks ritme van waarde leveren en afstemming van het team in de daily stand-up, zorgt voor focus op snelle levering.
- Het leveren van waarde voor de eindgebruiker staat centraal (user-stories).

#### 5 RESPECTEER MENSEN

De mensen die het werk doen hebben meer dan wie ook zicht op de juistheid van de aanpak en de resultaten. Door een team zelf verantwoordelijk te maken voor de eigen processen en deze zelf te laten verbeteren, worden leerpunten eenvoudiger en sneller verwerkt. Beslissingsprocessen worden daardoor beter en sneller.



#### RESPECTEER MENSEN

- Mensen staan centraal: zelfsturende stabiele crossfunctionele teams.
- Het team bepaalt hoeveel werk het aan kan via de sprintbacklog (pull).
- Het team geeft zelf aan wat het betekent wanneer het 'klaar' is (definition-of-done).
- Het team is zelf verantwoordelijk voor de aanpak en procesverbeteringen.

#### 6 BOUW KWALITEIT IN

Kwaliteit achteraf proberen in te brengen, bijvoorbeeld door aan het einde te testen, is zeer inefficiënt. Daarom moet het leveren van kwaliteit en borgen van kwaliteit centraal staan in alles wat het team doet. Van iedere afzonderlijke stap in het proces moet het resultaat al getest zijn. Alleen dan kan er snel worden geleverd. Ook hiermee wordt verspilling uit het systeem gehaald. Werk dat namelijk niet tot kwaliteit leidt, is verspilling.



#### BOUW KWALITEIT IN

- Dagelijkse feedback op kwaliteit door continu testen en leveren van werkende software.
- Continue feedback op kwaliteit door cycli van enkele weken (sprints).
- Zelfsturende teams van professionals die zelf verantwoordelijk zijn voor het eindresultaat en directe controle hebben op de kwaliteit van het werk.

#### 7 HOUD ZICHT OP HET GEHEEL

Het grote risico van complex werk is dat het zodanig wordt opgedeeld in losse delen dat die, ieder voor zich, prima te overzien zijn maar het zicht op het geheel vervaagt. Daarom moet er in alles wat er gebeurt altijd een compleet kader zijn waarbinnen het voor iedereen volstrekt helder is hoe de onderdelen bijdragen aan het totaal.



#### HOUD ZICHT OP HET GEHEEL

- Het eindresultaat (werkende software) staat in alles centraal.
- De backlog is langer dan een sprint, het team weet daardoor wat de stip op de horizon is.
- Een crossfunctioneel team dat gezamenlijk het eindresultaat realiseert. Geen losse disciplines maar één gezamenlijk beeld en eindresultaat.

