

Een voorbeeldchecklist voor Scrum Masters

Michael James
(mj@seattlescrum.com)
14 September 2007
(Revised 2 Feb 2016)

<http://ScrumMasterChecklist.org>

Nederlandstalige vertaling: Frederik Vannieuwenhuyse

Een fulltime begeleider?

Een adequate Scrum Master kan twee of drie teams tegelijkertijd begeleiden. Wanneer je de rol gedeeltelijk invult, heb je de tijd beschikbaar voor het organiseren van vergaderingen, het afdwingen van timeboxen en het reageren op problemen die mensen expliciet aan jou komen melden. Vind je dit voldoende? Het team zal het dan hoogstwaarschijnlijk nog altijd redelijk goed doen vergeleken met de standaardverwachtingen van de organisatie (voor de invoering van Scrum). Er zal ook niks dramatisch gebeuren of veranderen.

Indien je een team vooropstelt dat op een geweldige manier doelen bereikt die niemand eerder voor mogelijk hield - en dit in een werkelijk veranderende organisatie - streef er dan naar om een geweldige Scrum Master te zijn.

Een geweldige Scrum Master is een Scrum Master van maximaal één team.

Onze aanbeveling is om bij de start één toegewijde Scrum Master te voorzien per team van ongeveer zeven personen.

Als Scrum Master, als je nog niet al het werk hebt ontdekt dat er mogelijks te doen is, stem dan af op je Product Owner, je team, de engineering praktijken van je team en de organisatie buiten je team. Hoewel er geen werkend recept is voor iedereen of elke situatie, zijn hieronder typische aspecten beschreven die Scrum Masters over het hoofd zien.

Aanbevolen werkwijze: Markeer elk vakje met √, Δ, ? of N.v.t., zoals beschreven op de laatste pagina.

Deel I -- Hoe gaat het met mijn Product Owner?

Scrum Masters verbeteren de effectiviteit van Product Owners door hen te helpen manieren te vinden om de Product Backlog en het release plan bij te houden. (Merk op dat de Product Owner verantwoordelijk is voor de geordende product backlog.)

- ☐ Is de Product Backlog geordend volgens de meest recente inzichten?
- ☐ Zijn eisen en wensen van alle belanghebbenden vastgelegd in de Product Backlog? Denk eraan: een backlog is continue in beweging en krijgt vorm volgens opkomende inzichten.
- ☐ Is de grootte van de Product Backlog beheersbaar? Om een beheersbaar aantal items te hebben, zijn items naar boven toe gedetailleerder, met algemenere items ("epics") meer onderaan in de backlog. Het biedt geen voordeel om items lager in de Product Backlog te over analyseren. Vereisten zullen sowieso veranderen door nieuwe inzichten die ontstaan door continue conversaties tussen belanghebbenden/klanten en het product in ontwikkeling.
- ☐ Welke vereisten (voornamelijk deze bovenaan de Product Backlog) kan je beter beschrijven als user stories die onafhankelijk, onderhandelbaar, waardevol, schatbaar, klein en testbaar zijn?
- ☐ Heb je jouw Product Owner uitgelegd wat technical debt is en hoe dit te voorkomen? Een mogelijkheid is om het voorzien van geautomatiseerde tests en refactoring op te nemen in de Definitie van "Done" voor elk backlog item.

- ☐ Biedt de backlog inzicht in informatie, rechtstreeks toegankelijk en zichtbaar voor alle belanghebbenden?
- ☐ Indien je een geautomatiseerd systeem gebruikt voor backlog beheer, weet iedereen op welke manier dit gemakkelijk te gebruiken is? Geautomatiseerde beheerssystemen vergroten het risico om informatie te verbergen, indien de Scrum Master deze informatie niet actief transparant maakt.
- ☐ Kan je helpen om informatie meer te verspreiden door iedereen een geprinte versie te tonen?
- ☐ Kan je helpen om informatie meer te verspreiden door het maken van grote zichtbare informatiegrafieken?
- ☐ Heb je jouw Product Owner al geholpen met het organiseren van backlog items in gepaste release of prioriteitsgroepen?
- ☐ Is iedereen op de hoogte of het release plan nog realistisch is? Je kan iedereen Product/Release Burndown grafieken tonen nadat items als “done” beschouwd zijn tijdens elke Sprint Review. Grafieken tonen de vooruitgang van effectief opleverde Product Backlog items. Nieuwe items die toegevoegd zijn maken wijzigingen in scope en planning zichtbaar.
- ☐ Heeft jouw Product Owner het release plan aangepast na de laatste Sprint Review? Product Owners die op tijd voldoende geteste producten opleveren, herplannen de release elke sprint.

Deel II -- Hoe gaat het met mijn team?

We moedigen zeker aan om het goede voorbeeld te geven wat betreft onderlinge samenwerking met teamleden. Er is wel een risico dat je verloren geraakt in technische taken. Denk na over jouw belangrijkste verantwoordelijkheden ten opzichte van het team:

- ☐ Is jouw team in een toestand van *flow*? Enkele kenmerken van deze toestand:¹
 - Duidelijke doelen (verwachtingen en afspraken zijn waarneembaar en doelen zijn haalbaar, passend bij iemands vaardigheden en capaciteiten).
 - Concentratie en focus, doelgerichtheid door hoge concentratie op een beperkt aandachtsgebied.
 - Een verlies van zelfbewustzijn waarbij men zich geheel focust op de bezigheid en zichzelf vergeet (het samensmelten van actie en bewustzijn).
 - Directe en onmiddellijke feedback (succes en falen zijn direct duidelijk, zodat men daarop het eigen handelen en gedrag direct kan aanpassen).
 - Een gevoel van uitdaging waarbij het wel haalbaar is (een evenwicht: niet te moeilijke, maar ook niet te makkelijke bezigheden)
 - Het gevoel van persoonlijke controle over de situatie en/of activiteit.
 - De bezigheid is intrinsiek de moeite waard en belonend, het voelt niet als een last.
- ☐ Kunnen teamleden het met mekaar vinden, samen plezier hebben en elkaars succes te vieren?
- ☐ Houden teamleden elkaar verantwoordelijk voor hoge normen en dagen ze elkaar uit om te groeien?
- ☐ Zijn er problemen/onbenutte opportuniteiten die het team niet bespreekt omdat deze te oncomfortabel zijn?²

¹ Mihaly Csikszentmihalyi, *Flow: The Psychology of Optimal Experience* (1990).

² Marshall Rosenberg, *Nonviolent Communication: A Language of Life: Life-Changing Tools for Healthy Relationships* (2003). Overweeg ook om een professionele begeleider in te schakelen die ongemakkelijke gesprekken comfortabeler kan maken.

- ☐ Heb je verschillende werkvormen en locaties geprobeerd voor Sprint Retrospectives?³
- ☐ Focust het team zich op het Sprint doel? Je kan halverwege de Sprint een check-up faciliteren om de acceptatie criteria van de Product Backlog Items gepland voor de Sprint opnieuw te bekijken.
- ☐ Weerspiegelt het Sprint-taakbord wat het team doet? Let op voor "verborgen werk" van niet zichtbare taken en taken die langer duren dan een dag werk. Taken die ongerelateerd met de Sprint en het Sprint doel, vormen een belemmering voor de betrokkenheid en focus van het team.
- ☐ Bestaat het team uit 3-9 mensen met een voldoende mix van vaardigheden om een potentieel leverbaar product increment te creëren?
- ☐ Is het takenbord van het team up-to-date?
- ☐ Zijn de artefacten m.b.t. zelfmanagement zichtbaar voor het team, en handig voor het team om te gebruiken?
- ☐ Zijn deze artefacten voldoende beschermd tegen inmenging van buitenaf? Teveel controle van de dagelijkse activiteiten door mensen buiten het team kan de interne transparantie en zelfsturing van het team belemmeren.
- ☐ Bieden teamleden zich vrijwillig aan voor taken?
- ☐ Is de nood aan aflossing van technische schuld expliciet gemaakt in de definitie van 'done', waardoor de code geleidelijk verbetert en aangenamer wordt om in te werken?
- ☐ Laten teamleden hun functietitel achterwege in het team? Is er een gezamenlijke verantwoordelijkheid voor alle aspecten van het overeengekomen werk (testen, gebruikersdocumentatie, enz.)?

Deel III -- Hoe gaat het met onze engineering praktijken?

- ☐ Heeft het systeem in ontwikkeling een "push-to-test"-knop waarmee iedereen (het team zelf of een ander team) gemakkelijk kan detecteren wanneer ze een regressiefout hebben veroorzaakt (voorheen werkende functionaliteit die nu gebroken is)? Meestal wordt dit bereikt via het xUnit-framework (JUnit, NUnit, enz.).
- ☐ Is er een goede balans tussen geautomatiseerde end-to-end systeemtests (ook wel "functionele tests" genoemd) en geautomatiseerde unit-tests?
- ☐ Schrijft het team zowel systeemtests als unit tests in dezelfde taal als het systeem dat ze ontwikkelen? De samenwerking wordt er niet beter op door het gebruik van propriëtaire scripttalen of tools waarvan alleen een deel van het team weet hoe deze te onderhouden.
- ☐ Heeft het team de nuttige grijze zone tussen systeemtests en unittests ontdekt?⁴
- ☐ Slaat een continue integratieserver⁵ automatisch alarm als iemand een regressiefout veroorzaakt? Kan deze feedback lus teruggebracht worden tot uren of minuten? ("*Dagelijkse builds zijn voor watjes.*" - Kent Beck)
- ☐ Toont de continue integratieserver het resultaat van *alle* tests?

³ Derby/Larson *Agile Retrospectives: Making Good Teams Great* (2006).

⁴ <http://blogs.collab.net/agile/junit-is-not-just-for-unit-testing-anymore>

⁵ <http://www.martinfowler.com/articles/continuousIntegration.html>

- ☐ Hebben teamleden het plezier ontdekt van het continu ontwerpen en constante refactoring⁶, als alternatief voor "Big Up Front Design"? Refactoring heeft een strikte definitie: het veranderen van de interne structuur zonder het externe gedrag te veranderen. Refactoring moet meerdere keren per uur plaatsvinden, telkens wanneer er dubbele code is, complexe voorwaardelijke logica (zichtbaar door te veel inspruing of lange methodes in code), slecht benoemde identifiers, overmatige koppeling tussen objecten, enz. Refactoring toepassen met vertrouwen is alleen mogelijk met behulp van geautomatiseerde testing. Het laten links liggen van refactoring maakt het moeilijk om het product in de toekomst te veranderen; voornamelijk omdat het moeilijk is om goede ontwikkelaars te vinden die aan code willen werken die slecht onderhouden is.
- ☐ Omvat de Definitie van "Done" voor elk Product Backlog item volledig geautomatiseerde testing en refactoring? Het leren en toepassen van Test-Driven Development (TDD) maakt dit realistischer.
- ☐ Programmeren teamleden meestal per twee? Pair-programmering kan de onderhoudbaarheid van code drastisch verhogen en het aantal bugs verlagen. Het daagt de grenzen van mensen uit. Soms lijkt pair-programmering langer te duren (wanneer we meten volgens regels code in plaats van opgeleverde functionaliteit). Geef het goede voorbeeld door het initiëren van werkdagen met pairing. Bij sommige teamleden zal deze manier van werken de voorkeur genieten.

Deel IV -- Hoe gaat het met de organisatie?

- ☐ Vindt er voldoende communicatie tussen teams plaats? "*Scrum of Scrums*" is slechts één manier om dit te bereiken en zelden de beste.⁷
- ☐ Zijn teams in staat om zelfstandig werkende features te produceren, zelfs over architecturale grenzen heen?⁸
- ☐ Zijn Scrum Masters met elkaar in gesprek en werken ze aan de lijst van organisatorische belemmeringen?
- ☐ Worden de organisatorische belemmeringen waar nodig op de muur van het kantoor van de directeur gekleefd, zodat deze volledig zichtbaar zijn? Kunnen de kosten worden gekwantificeerd in Euros, verlies in tijd om iets op de markt te brengen, verloren kwaliteit of andere gemiste opportuniteiten? (Leer wel uit de fouten van Ken Schwaber: "*Een dode Scrum Master is een nutteloze Scrum Master.*"⁹)
- ☐ Is jouw organisatie één van de weinige met loopbaanmogelijkheden die compatibel zijn met teams die collectieve doelen stellen? Antwoord 'nee' als er een carrière-beloning¹⁰ bestaat om programmeer- of architectuurwerk te doen ten koste van testen, testautomatisering of gebruikersdocumentatie.
- ☐ Is jouw organisatie door de vakpers of andere onafhankelijke bronnen erkend als één van de beste werkplekken of als leider in jouw sector of branche?
- ☐ Creëer je een *lerende organisatie*?

⁶ Martin Fowler, *Refactoring: Improving the Design of Existing Code* (1999).

⁷ See <http://less.works/less/framework/coordination-and-integration.html> for alternatives.

⁸ <http://FeatureTeamPrimer.org/>

⁹ Ken Schwaber, *Agile Project Management with Scrum* (2004)

¹⁰ Alfie Kohn, *Punished By Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes* (1999)

Conclusie

Als je de meeste van deze items kunt afvinken en je hebt nog tijd over tijdens de dag, dan hoor ik graag van je.

Er is geen standaardformule om menselijke vindingrijkheid te laten ontspruiten. Dit document somt de punten op die in jouw situatie wel of niet relevant zijn.

Als je eenmaal begint te beseffen wat je allemaal zou kunnen doen om een verschil te maken, heb je misschien een terughoudendheid om het daadwerkelijk te doen. Dit is een teken dat je op de goede weg bent.

FORMULIER ORGANISATIEBELEMNERING

PROBLEEM:

HOOFDORZAAK (GEBRUIK VIJF KEER "WAAROM?"):

BEDRIJFSIMPACT:

EMOTIONELE IMPACT:

DUIDELIJK, UITVOERBAAR VERZOEK:

FORMULIER ORGANISATIEBELEMNERING

PROBLEEM:

HOOFDORZAAK (GEBRUIK VIJF KEER "WAAROM?"):

BEDRIJFSIMPACT:

EMOTIONELE IMPACT:

DUIDELIJK, UITVOERBAAR VERZOEK:

FORMULIER ORGANISATIEBELEMNERING

PROBLEEM:

HOOFDORZAAK (GEBRUIK VIJF KEER "WAAROM?"):

BEDRIJFSIMPACT:

EMOTIONELE IMPACT:

DUIDELIJK, UITVOERBAAR VERZOEK:

FORMULIER ORGANISATIEBELEMNERING

PROBLEEM:

HOOFDORZAAK (GEBRUIK VIJF KEER "WAAROM?"):

BEDRIJFSIMPACT:

EMOTIONELE IMPACT:

DUIDELIJK, UITVOERBAAR VERZOEK:

FORMULIER ORGANISATIEBELEMNERING

PROBLEEM:

HOOFDORZAAK (GEBRUIK VIJF KEER "WAAROM?"):

BEDRIJFSIMPACT:

EMOTIONELE IMPACT:

DUIDELIJK, UITVOERBAAR VERZOEK:

FORMULIER ORGANISATIEBELEMNERING

PROBLEEM:

HOOFDORZAAK (GEBRUIK VIJF KEER "WAAROM?"):

BEDRIJFSIMPACT:

EMOTIONELE IMPACT:

DUIDELIJK, UITVOERBAAR VERZOEK:

INSTRUCTIES

Als je deze checklist als opdracht hebt gekregen en je huidige (of meest recente) werkgever heeft iets als Scrum geprobeerd, pas dit dan toe op wat je daar hebt gezien. Markeer elk item met één van de volgende:

- √ ("goed genoeg")
- Δ ("kan worden verbeterd en ik weet hoe te beginnen")
- ? ("kan worden verbeterd, maar hoe is nog niet duidelijk")
- N.v.t. ("niet van toepassing" of "levert geen voordeel op")

Of, als jouw huidige (of meest recente) werkgever nog nooit zoiets als Scrum heeft geprobeerd, markeer dan elk item met één van de volgende:

- √ ("goed bezig" of "geen problemen om dit goed te doen")
- Δ ("lijkt een uitdaging en ik weet hoe ik moet beginnen")
- ? ("lijkt een uitdaging en ik weet niet hoe ik moet beginnen")
- N.v.t. ("niet van toepassing" of "zou geen voordeel opleveren")

Wanneer je alle items in de lijst hebt aangeduid, noteer dan 2-6 organisatorische belemmeringen aan op de bijgevoegde formulieren, ongeacht of deze al dan niet zijn afgeleid uit deze checklist. Kies belemmeringen waarvan je ten minste 1% hoop hebt om iets veranderen.