

Un exemplu de listă de verificare pentru Scrum Masters

Autor: Michael James (mj@seattlescrum.com)

Creat la data de 14 februarie 2017 și revizuit pe data de 8 februarie 2023

Tradus din limba engleză (<https://scrummasterchecklist.org>) la data de 10 iulie 2024

Traducător: [Madlen McCrea | LinkedIn](#)

Un facilitator cu normă întreagă?

Un Scrum Master calificat poate lucra cu două sau trei echipe Scrum în același timp. Dacă ești mulțumit / mulțumită să îți limitezi rolul la organizarea ședințelor, asigurându-te că ședințele se termină la timp, și de asemenea rezolvând obstacolele raportate de echipa Scrum, atunci poți face față rolului de Scrum Master în capacitate redusă. Nu se va întâmpla nimic catastrofal.

Poți să fii *bun* în rolul de Scrum Master dacă vrei ca echipa Scrum să fie o echipă de succes care realizează lucruri pe care oamenii din echipă nu au știut că le pot îndeplini.

O persoană în rolul de Scrum Master poate fi *de succes* dacă lucrează doar cu o echipă Scrum, în special când echipa Scrum este formată din șase oameni și începe să lucreze pentru o inițiativă nouă.

Dacă în rolul de Scrum Master, nu ai descoperit încă toate activitățile care trebuie să fie îndeplinite, atunci lucrează direct cu persona în rolul de Product Owner și cu echipa Scrum, și uită-te la practicile echipei de ingineri, și ale companiei. Deși nu există doar o rețetă generală pentru toată lumea, am subliniat câteva lucruri tipice pe care le-am observat să fie trecute cu vederea de cei în rolul de Scrum Master. Vă rog să marcați fiecare pătrat cu ✓, Δ, ?, sau N/A, pe baza explicațiilor pe care le găsiți în ultima pagină a acestui document.

Partea I – Ce face persoana în rolul de Product Owner?

Persoanele în rolul de Scrum Master îmbunătățesc în multe feluri eficiența persoanei în rolul de Product Owner.

- ☐ Indiferent de numărul de echipe Scrum, o singură persoană este numită în rolul de Product Owner pentru un produs?
- ☐ Prioritățile produsului din lista numită Product Backlog sunt stabilite recent de persoana în rolul de Product Owner?
- ☐ Dorințele și cerințele noi pentru produs sunt incluse în lista numită Product Backlog? Notă: Această listă Product Backlog este *emergentă*.
- ☐ Conținutul listei numită Product Backlog este ușor de folosit? Ține elementele de la începutul listei mai detaliate, cu tipul de epic general la sfârșitul listei. Nu este productiv să analizezi continuu elementele care nu sunt o prioritate, nefiind la începutul listei. Planurile se vor schimba pe baza discuțiilor dintre utilizatori/client și dezvoltarea produsului..
- ☐ Oricare dintre cerințe (în special cele de la începutul listei numită Product Backlog) poate fi exprimată mai bine ca independentă, negociabilă, valoroasă, estimabilă, simplă, și testabilă (INVEST User Story)¹?
- ☐ Persoana în rolul de Product Owner știe ce este datoria tehnică (vezi „technical debt”) și cum să o evite? O soluție ar fi să adaugi testarea automată și simplificarea codului de program (vezi „cod refactoring”) la fiecare element din listă atunci când se stabilește ce înseamnă complet (vezi „definition of done”).
- ☐ Lista numită Product Backlog este o *sursă de informare* și poate fi văzută imediat de toți cei interesați?
- ☐ În cazul în care lista numită Product Backlog este automatizată, știu toți din echipă cum să o folosească? Din păcate, automatizarea blochează colaborarea, devenind un „frigider cu informații” (vezi „information refrigerators”).

¹ <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

- ☐ Poți ajuta la difuzarea informațiilor prin a-i învăța pe alții cum să creeze diagrame vizibile?
- ☐ L-ai ajutat pe Product Owner să organizeze elementele din lista numită Product Backlog în funcție de priorități?
- ☐ Știe fiecare dacă planul de lansare numit Release Plan este realist? După ședința numită Sprint Review Meeting, la care elementele din listă au fost acceptate ca si „complete”, ar trebui să le arăți la toți progresul făcut folosind diagramele numite Product / Release Burndown Charts². Diagramele care reprezintă cât din produs, numit Product Backlog Increment (PBI), este completat cât și ce a fost adăugat, pot indica imediat orice abatere de la scop / plan (vezi „*schedule*”).
- ☐ Estimările de lansare au fost revizuite de Product Owner după ultima ședință numită Sprint Review Meeting? Dacă vrei să lansezi la o dată exactă un produs testat, atunci trebuie să reconsideri constant scopul. Asta înseamnă să amâni anumite elemente minore pentru o lansare viitoare atunci când sunt descoperite elemente mai importante. Anumite produse pot fi lansate la fiecare Sprint sau de mai multe ori în timpul unui Sprint.

Part II – Ce face echipa?

Persoana în rolul de Scrum Master este încurajată să fie un exemplu pentru echipa Scrum și să colaboreze cu echipa, chiar dacă există riscul să fie implicat prea mult în activitățile tehnice. Consideră responsabilitățile tale cu echipa Scrum:

- ☐ Echipa este într-o stare de *mișcare continuă* (vezi „*flow*”)? Câteva trăsături ale acestei stări³:
- Obiective clare (golurile și regulile sunt tangibile iar obiectivele se pot realiza, aliniindu-se în mod corespunzător cu setul de abilități disponibile).
 - Concentrare și atenție, un nivel înalt de concentrare și o atenție limitată.
 - O pierdere a sentimentului de conștiință de sine, o combinație între acțiune și conștiință.
 - Un răspuns direct și imediat (succesele și eșecurile sunt evidente, iar comportamentul se poate ajusta după cum este nevoie).
 - Un echilibru între nivelul de abilitate și provocare (o activitate nu este nici prea ușoară și nici prea grea).
 - Un sens de control personal sau de cunoaștere a activității de îndeplinit.
 - O muncă plină de satisfacții, naturală, fără efort.
- ☐ Oamenii din echipă lucrează cu plăcere împreună, glumesc între ei, și sărbătoresc succesele fiecăruia?
- ☐ Oamenii din echipă au standarde înalte și se stimulează unii pe alții să devină mai buni profesional?
- ☐ Oamenii din echipă nu discută nici o problemă sau oportunități pentru că este incomod?⁴
- ☐ Pentru discuțiile retrospective (vezi „Sprint Retrospectives”⁵) cu echipa ai încercat diferite metode și locuri?
- ☐ Echipa s-a concentrat pe obiectivele din Sprint? Consideră o reevaluare a planului (vezi „*Sprint Plan*”) la mijlocul perioadei.

² Mike Cohn, *Agile Estimation and Planning*. (2005).

³ Mihaly Csikszentmihalyi, *Flow: The Psychology of Optimal Experience* (1990).

⁴ Marshall Rosenberg, *Nonviolent Communication: A Language of Life: Life-Changing Tools for Healthy Relationships* (2003). Also consider enlisting a professional facilitator who can make uncomfortable conversations more comfortable.

⁵ Derby/Larson *Agile Retrospectives: Making Good Teams Great* (2006).

- ☐ Activitățile pe care echipa le completează sunt reflectate în lista Sprint Backlog? Fii atent / atentă la „materia întunecată” ce conține activitățile care nu sunt menționate sau cele care nu se completează într-o zi de lucru. Un impediment în Sprint este orice activitate care nu are nici o legătură cu munca ce trebuie completată în Sprint.
- ☐ Echipa ta este compusă din 3 până la 9 oameni care au toate cunoștințele necesare să producă un produs potențial de lansare?
- ☐ Tabela cu sarcinile echipei este actualizată?
- ☐ Documentele create de echipe sunt vizibile și ușor de folosit? Sunt protejate? Controlul excesiv al sarcinilor zilnice de către oameni care nu fac parte din echipă poate împiedica modul echipei de a lucra transparent și independent.
- ☐ Oamenii din echipă se oferă să lucreze anumite activități? Când am lucrat prima oară într-o echipă Scrum, ne-am simțit ca și voluntari plătiți. Ceva este greșit dacă nu ai acest sentiment.
- ☐ S-a stabilit explicit ”rambursarea datoriei tehnice” (vezi „*technical debt repayment*”) și ce este definit complet (vezi „*definition of done*”), făcând mai plăcută scrierea în cod al programului?
- ☐ Oamenii din echipă nu se folosesc de titlul pe care îl au și sunt responsabili ca și grup pentru toate activitățile necesare (testare, manualul utilizatorului, etc.)?

Part III – Cum sunt practicile noastre de inginerie?

- ☐ Sistemul în curs de dezvoltare, are un buton „apasă să testezi” ce îi permite oricui (din echipă sau din altă echipă) să descopere ușor când au provocat o eroare pentru că funcționalitatea acestuia s-a stricat)? De obicei se folosește xUnit (JUnit, NUnit, etc.).
- ☐ În cazul testării automate, testele automate pentru verificarea întreagă a funcționării sistemului (vezi „end-to-end system tests” a.k.a. "funcțional tests") sunt într-un echilibru cu testele automate pentru testarea executării programului (vezi „unit tests”)?
- ☐ Echipa creează în același limbaj de programare testele pentru funcționarea sistemului și testele pentru executarea programului? Colaborarea între membrii echipei nu se îmbunătățește dacă se folosesc limbajele de programare cu cod special sau programele specializate pe care doar câțiva oameni din echipă le cunosc și știu cum să le mențină.
- ☐ Echipa a descoperit „zona gri” (nici albă și nici neagră) dintre testele folosite pentru verificarea funcționării sistemului și testele folosite pentru depistarea erorilor de programare?
- ☐ Calculatorul folosit pentru integrare continuă, numit „continuous integration⁶ server”, sună o alarmă automat atunci când o persoană creează o eroare în timpul testării funcționalității existente a sistemului (vezi „regression testing”) ? Se poate reduce acest test la ore sau minute? („Novicii sunt cei care compilează codul de program zilnic.” / "Daily builds are for wimps." -- Kent Beck)
- ☐ Toate testele automate sunt incluse în testarea de integrare continuă (vezi „continuous integration”)?
- ☐ Oamenii din echipă execută constant proiectarea și factorizarea codului de program (vezi „continuous design and constant refactoring”⁷), ca și alternativă la proiectarea inițială a întregului sistem („Big Up Front Design”) ? Definiția pentru factorizare (vezi „refactoring”) este strictă: Modul în care aplicația funcționează nu se schimbă atunci când se face modifică sau se reorganizează codul programului. Factorizarea trebuie să fie executată constant sau cel puțin de câteva ori pe oră, fie când același cod de program este folosit de mai multe ori, fie când logica programului are condiții complicate ce pot fi văzute prin indentarea

⁶ <http://www.martinfowler.com/articles/continuousIntegration.html>

⁷ Martin Fowler, *Refactoring: Improving the Design of Existing Code* (1999).

codului de program, fie când funcțiile în program nu au nume explicit, fie când obiectele din cod sunt cuplate excesiv (vezi „object programming”), ș.a.m.d.. Factorizarea poate fi făcută cu încredere atunci când sunt utilizate testelor automate care testează întregul sistem. Modificarea ulterioară a produsului va fi mult mai complicată dacă se ignoră factorizarea pentru că nu este ușor să găsești programatori buni care vor fi dispuși să modifice un cod scris prost.

- ☐ Testarea automată a întregului sistem este inclusă în definiția pentru element „completat” (vezi „definition of done”) pentru fiecare element din lista numită Product Backlog? Probabilitatea de a reuși acest lucru este mărită dacă se învață programarea folosind testarea (vezi „Test Driven Development (TDD)”).
- ☐ Membrii echipei programează în pereche (vezi „pair programming”) în majoritatea timpului? Chiar dacă limitele oamenilor sunt contestate și aparent durează mai mult (mai ales dacă performanța este determinată de numărul de linii de cod în locul unui produs funcțional), în acest fel de lucru se reduce efortul de a modifica codul de program cât și numărul de defecte create. Fii un exemplu pentru echipă și inițiază zilele de lucru în pereche (vezi „pair programming”) sau sesiunile în care toată echipa completează aceeași activitate împreună și în același timp, la un singur calculator (vezi „mobbing sessions”). Unii dintre ei vor prefera să lucreze în felul ăsta.

Part IV – Care este eficiența în organizație?

- ☐ Echipele din organizație comunică eficient între ele? O metodă pentru a realiza acest lucru, dar rareori cea mai bună, este participarea la ședințele cu toate persoanele care sunt în rolul de Scrum Master (vezi „Scrum of Scrums”⁸).
- ☐ Echipele pot produce funcționalitate ce poate fi lansată, chiar și peste limitele arhitecturale⁹?
- ☐ Conduci o retrospectivă generală (vezi „Overall Retrospective”¹⁰) pentru a rezolva problemele echipelor din organizație, atât organizațional cât și sistemice?
- ☐ Când e cazul, impedimentele companiei sunt vizibile pe peretele biroului directorului de programare? Prețul întârzierilor poate fi calculat în valoare monetară, ori în timp de lansare pierdut, sau prin calitate insuficientă, sau prin numărul de clienți pierduți? (Învață din greșelile lui Ken Schwaber: un Scrum Master care nu este activ este un Scrum Master inutil. Vezi "A dead Scrum Master is a useless Scrum Master."¹¹).
- ☐ Organizația dumneavoastră este una dintre puținele care au carierele oamenilor compatibile cu obiectivele colective ale echipelor dumneavoastră? Răspunde „nu” dacă există un stimulent în carieră (vezi „career incentive”¹²) pentru a efectua lucrări de programare sau arhitectură în detrimentul testării ori a folosirii testelor automate sau de a crea documentația utilizatorului.
- ☐ Organizația dumneavoastră a fost recunoscută de presa de specialitate sau de alte surse independente drept una dintre cele mai bune locuri de muncă sau ca și lider în industria dumneavoastră?
- ☐ Creezi o organizație în care se învață (vezi „learning organization”)?

Concluzie

⁸ See <https://less.works/less/framework/coordination-and-integration/> for alternatives.

⁹ <https://less.works/less/structure/feature-teams>

¹⁰ <https://less.works/less/framework/overall-retrospective/>

¹¹ Ken Schwaber, *Agile Project Management with Scrum* (2004)

¹² Alfie Kohn, *Punished By Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes* (1999)

Contactează-mă dacă marchezi cu bine majoritatea elementelor din această listă și mai ai timp liber în timpul zilei.

Ingenuitatea umană nu se dezvoltă folosind o formulă existentă. Această listă conține puncte de vedere care s-ar putea să te ajute. În momentul în care îți dai seama ce ai putea face pentru ca să faci o diferență, s-ar putea să îți fie frică să o faci iar acesta este un semn că ești pe drumul cel bun.

Obstacol Organizațional

Problemă:

Cauză (folosește metoda „De ce?” de cinci ori”, vezi „five times “Why?””):

Impact Economic (afacere, activitate, operațiune):

Impact Emoțional:

Cerere pentru suport (clară și acționabilă):

Obstacol Organizațional

Problemă:

Cauză (folosește metoda „De ce?” de cinci ori”, vezi „five times “Why?””):

Impact Economic (afacere, activitate, operațiune):

Impact Emoțional:

Cerere pentru suport (clară și acționabilă):

Obstacol Organizațional

Problemă:

Cauză (folosește metoda „De ce?” de cinci ori”, vezi „five times “Why?””):

Impact Economic (afacere, activitate, operațiune):

Impact Emoțional:

Cerere pentru suport (clară și acționabilă):

Obstacol Organizațional

Problemă:

Cauză (folosește metoda „De ce?” de cinci ori”, vezi „five times “Why?””):

Impact Economic (afacere, activitate, operațiune):

Impact Emoțional:

Cerere pentru suport (clară și acționabilă):

Obstacol Organizațional

Problemă:

Cauză (folosește metoda „De ce?” de cinci ori”, vezi „five times “Why?””):

Impact Economic (afacere, activitate, operațiune):

Impact Emoțional:

Cerere pentru suport (clară și acționabilă):

Obstacol Organizațional

Problemă:

Cauză (folosește metoda „De ce?” de cinci ori”, vezi „five times “Why?””):

Impact Economic (afacere, activitate, operațiune):

Impact Emoțional:

Cerere pentru suport (clară și acționabilă):

INSTRUCȚIUNI

Folosește această listă de verificare dacă angajatorul dumneavoastră actual (sau cel mai recent) încearcă să folosească ceva de genul Scrum și ai primit-o ca sarcină de pregătire. Marchează fiecare punct cu una dintre următoarele:

✓ (pentru „corect, fac bine”)

Δ (pentru „se poate îmbunătății și știu cum”)

? (pentru „se poate îmbunătății, dar cum?”)

N/A (pentru „nu este cazul” sau „nu ar oferi nici un beneficiu”)

Sau, dacă angajatorul dumneavoastră actual (sau cel mai recent) nu a încercat nimic ca Scrum, atunci marchează fiecare element din listă cu unul dintre următoarele:

✓ (pentru „corect, fac bine” sau „ar fi ușor să fac bine”)

Δ (pentru „ar fi o provocare și știu cum să încep”)

? (pentru „ar fi o provocare și nu știu cum să încep”)

N/A (pentru „nu este cazul” sau „nu ar oferi nici un beneficiu”)

După ce ai terminat de evaluat lista de verificate și ai marcat toate punctele din listă, folosește formularele Obstacol Organizațional atașate și declară două până la șase obstacole organizaționale, chiar dacă sunt sau nu derivate din această listă. Alege impedimentele pe care speri să le schimbi cu o șansă de cel puțin 1%.