

Un esempio di check-list per Scrum Masters

Michael James
(mj@seattlescrum.com)
Rivista 8/2/2022

<https://scrummasterchecklist.org>

Traduzione a cura di Francesco Tatullo (www.linkedin.com/in/francescotatullo)

Un facilitatore a tempo pieno?

Uno Scrum Master adeguato può gestire due o tre team alla volta. Se vi accontentate di limitare il vostro ruolo all'organizzare meeting, far rispettare le timebox e rispondere agli impedimenti che vi segnalano esplicitamente, ve la potete cavare dando a questo ruolo una attenzione part-time. Probabilmente non succederà nulla di catastrofico.

Ma se immaginate un team che ha successo in cose che prima non pensava neppure di poter fare, prendete in considerazione l'idea di essere un *ottimo* Scrum Master.

Un ottimo Scrum Master può gestire *un solo* team alla volta.

All'inizio noi raccomandiamo uno Scrum Master dedicato per un team di circa sei membri.

Se non avete ancora scoperto tutto il lavoro che c'è da fare, concentratevi sul vostro Product Owner, sul vostro team, sulle pratiche ingegneristiche del team, e sull'organizzazione al di là del team. Sebbene non ci siano prescrizioni valide per tutti, ho riportato le tipiche cose che ho visto venire trascurate dagli Scrum Master. Per favore contrassegnate ogni casella con √, Δ, ?, o N/A, come descritto nell'ultima pagina.

Parte I - Come va il mio Product Owner?

Gli Scrum Master migliorano l'efficacia del Product Owner in molti modi.

- A prescindere dal numero di team, c'è un Product Owner (e uno soltanto¹) per il prodotto?
- Il Product Backlog è ordinato per priorità secondo la sua visione più recente?
- Il Product Backlog contiene i requisiti e i desiderata emersi recentemente? Ricordate: il backlog *emerge*.
- Il Product Backlog è di dimensioni gestibili? Per mantenere un numero gestibile di elementi, mantenete le cose più granulari verso l'alto, con le epiche più generali in basso. È controproducente analizzare eccessivamente sotto la cima del Product Backlog. I piani cambieranno nel corso di uno scambio continuo tra il prodotto che si sviluppa e gli utenti/clienti.
- Ci sono requisiti (soprattutto verso la cima del Product Backlog) che potrebbero essere espressi meglio come user stories indipendenti, negoziabili, di valore, stimabili, piccole e testabili??
- Avete formato il vostro Product Owner sul debito tecnico e come evitarlo? Un pezzo del puzzle potrebbe essere aggiungere test automatici e refactoring nella "Definizione di Fatto" per ciascun elemento del backlog.

² <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

- Il backlog è un *radiatore di informazione*, visibile a tutti gli interessati?
- Se usate un tool automatico per la gestione del backlog, sanno tutti usarlo facilmente? Purtroppo, i tool di gestione automatica spesso diventano *refrigeratori di informazione*, di ostacolo alla collaborazione.
- Potete aiutare a condividere informazione insegnando ad altri come creare grandi diagrammi ben visibili?
- Avete aiutato il vostro Product Owner a raggruppare appropriatamente gli elementi del backlog in release, o gruppi prioritari?
- Sanno tutti se il release plan è ancora realistico? Potreste provare a mostrare a tutti il Burndown Chart per la release o il prodotto dopo che tutti gli elementi sono stati accettati come “finiti” durante ogni Sprint Review. I grafici che mostrano l’andamento sia degli elementi del backlog effettivamente finiti sia di quelli nuovi permettono di scoprire anticipatamente scostamenti di scopo o tempistiche.
- Il vostro Product Owner ha aggiornato le previsioni per la release dopo l’ultima Sprint Review? Se volete rilasciare un prodotto adeguatamente testato entro una certa data, dovete costantemente rivalutare lo scopo. Questo probabilmente richiede di posticipare certe cose ad una release successiva mano a mano che cose più importanti vengono individuate. Alcuni prodotti possono essere rilasciati ad ogni Sprint, o più volte durante uno Sprint.

Parte II – Come va il mio team?

Sebbene lo Scrum Master sia incoraggiato a dare l’esempio nel collaborare con i membri del team, c’è il rischio di perdersi in compiti tecnici. Considerate le vostre responsabilità principali verso il team:

- Il vostro team è in uno stato di *flusso*? Alcune caratteristiche di questo stato sono:
 - Obiettivi chiari (le aspettative e le regole sono individuabili e gli obiettivi raggiungibili, opportunamente allineati con le competenze e le abilità di ciascuno).
 - Concentrazione e focus, un alto livello di concentrazione su un campo di attenzione limitato.
 - Una perdita del senso del sé, una fusione tra azione e coscienza.
 - Feedback diretto e immediato (i successi e gli insuccessi nel corso dell’attività sono evidenti, così che il comportamento possa essere regolato come necessario).
 - Equilibrio tra il livello di abilità e difficoltà (l’attività non è troppo facile né troppo difficile).
 - Un senso di controllo personale sulla situazione o attività.
 - L’attività è intrinsecamente gratificante, quindi l’azione è compiuta senza sforzo.
- I membri del team danno l’impressione di piacersi l’un l’altro, si divertono assieme, e festeggiano i rispettivi successi?
- I membri del team si aspettano molto gli uni dagli altri, e si spronano a vicenda a migliorare?
- Ci sono problemi e opportunità di cui il team non discute perché si sente troppo a disagio?⁵

³ Mike Cohn, *Agile Estimation and Planning*. (2005).

⁴ Mihaly Csikszentmihalyi, *Flow: The Psychology of Optimal Experience* (1990).

⁵ 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.

- Avete provato una varietà di format e luoghi per le Retrospective? ⁶
- Il team ha mantenuto il focus sui goal dello Sprint? Prendete in considerazione un tagliando a metà Sprint per rivalutare lo Sprint Plan.
- La bacheca dello Sprint riflette quello che il team sta effettivamente facendo? Fate attenzione alla “materia oscura” dei task non esplicitati e ai task più grandi di una giornata di lavoro. Attività non collegate allo Sprint sono un intralcio allo Sprint.
- Il vostro team ha 3-9 persone con un mix di competenze sufficiente per creare un incremento del prodotto che possa essere potenzialmente rilasciato?
- La bacheca del vostro team è aggiornata?
- Gli artefatti per la auto-organizzazione del team sono visibili e comodi per il team da usare?
- Questi artefatti sono adeguatamente protetti da intrusi? Un eccesso di esame delle attività quotidiane da parte di persone esterne al team può ostacolare la trasparenza interna e l’auto-organizzazione del team.
- I membri del team si offrono volontari per particolari task? Nel mio primo team Scrum ci sentivamo come volontari che venivano anche pagati. Se non c’è quella sensazione, qualcosa da qualche parte è andata storta.
- La necessità di ripagare il debito tecnico è stata resa esplicita nella “Definizione di Fatto”, rendendo gradualmente il codice un “luogo di lavoro” più piacevole?
- I membri del team lasciano i loro titoli fuori dalla porta, e sono collettivamente responsabili per tutti gli aspetti del lavoro concordato (testing, documentazione per l’utente, ecc.)?

Parte III – Come vanno le pratiche ingegneristiche?

- Il Sistema che state sviluppando ha un pulsante “premi per testare” che permette a chiunque (nello stesso team o in un altro team) di accorgersi facilmente di aver causato una regressione (“rotto” una funzionalità che prima funzionava correttamente)? Di solito questo viene ottenuto usando un framework di tipo xUnit (JUnit, NUnit, ecc.)
 - Avete un equilibrio adeguato tra test automatici end-to-end (detti anche “test funzionali”) e unit tests?
 - Il team sta scrivendo sia i test di sistema che gli unit test nello stesso linguaggio usato per il sistema che sta sviluppando? La collaborazione non viene facilitata da linguaggi di scripting proprietari o strumenti di registrazione e play-back che soltanto alcuni nel team sanno mantenere.
 - Il vostro team ha scoperto quell’utile zona grigia tra test di sistema e unit test?
 - C’è un server di continuous integration⁷ che automaticamente dà l’allarme quando qualcuno causa una regressione? Si può ridurre la tempistica di questo feedback a qualche ora, o qualche minuto? (“Le build giornaliere sono per i deboli.” – Kent Beck)
-
- I risultati del server di continuous integration includono *tutti* i test?

⁶ Derby/Larson *Agile Retrospectives: Making Good Teams Great* (2006).

⁷ <http://www.martinfowler.com/articles/continuousIntegration.html>

- I membri del team hanno scoperto il piacere della progettazione continua e del refactoring costante⁸, come alternativa alla Grande Progettazione Iniziale? Il “Refactoring” ha una definizione ben precisa: modificare la struttura interna senza modificare il comportamento esterno. Il refactoring dovrebbe avvenire costantemente, o almeno diverse volte all’ora. Fate refactoring tutte le volte che c’è codice duplicato, logica condizionale complessa (individuabile dall’eccesso di indentazione o dalla presenza di metodi lunghi), identificatori con nomi inadeguati, accoppiamento eccessivo tra oggetti, ecc. Fare refactoring in sicurezza è possibile soltanto con la copertura dei test automatici. Trascurare il refactoring rende difficile cambiare il prodotto in futuro, specialmente visto che è difficile trovare bravi sviluppatori che vogliono lavorare su un cattivo codice.
- La “Definizione di Fatto” per ogni elemento del Product Backlog include la piena copertura con test automatici e il refactoring? Imparare il Test Driven Development (TDD) aumenta la probabilità di arrivarci.
- I membri del team programmano in due, o in gruppo, per la maggior parte del tempo? Questo può drasticamente aumentare la manutenibilità del codice e ridurre il tasso di difetti. Mette alla prova i limiti delle persone e qualche volta sembra che richieda più tempo (se misuriamo usando il numero di linee di codice invece delle funzionalità rilasciabili). Date l’esempio dando il via a giornate di lavoro in coppia, o sessioni di gruppo con i membri del team. Alcuni di loro cominceranno a preferire lavorare in questo modo.

Parte IV – Come va l’organizzazione?

- C’è la giusta quantità di comunicazione tra team? “Scrum di Scrums” è solo un modo di arrivarci, e raramente il migliore⁹.
- I team sono in grado di produrre funzionalità che funzionano, al di là dei confini architetturali?¹⁰
- State facendo le *Retrospective Globali*¹¹ per risolvere i problemi tra team, quelli organizzativi e sistemici all’interno dell’organizzazione?
- Dove opportuno, gli ostacoli organizzativi vengono affissi al muro dell’ufficio del capo dello sviluppo? I costi possono essere quantificati in dollari, time to market sprecato, qualità persa, o opportunità mancate con i clienti? (Ma imparate dagli errori di Ken Schwaber: “Uno Scrum Master morto è uno Scrum Master inutile.”¹²)
- La vostra organizzazione è una delle poche con percorsi di carriera compatibili con gli obiettivi collettivi dei vostri team? Rispondete “no” se c’è un incentivo¹³ di carriera a programmare o fare lavoro architetturale a spese del testing, della automazione dei test, o della documentazione utente.
- La vostra organizzazione è stata riconosciuta dalla stampa di settore o da altre fonti indipendenti come uno dei migliori luoghi di lavoro, o un leader nel vostro settore?
- State creando una *organizzazione che apprende*?

⁸ Martin Fowler, *Refactoring: Improving the Design of Existing Code* (1999).

⁹ See <https://less.works/less/framework/coordination-and-integration/> for alternatives.

¹⁰ <https://less.works/less/structure/feature-teams>

¹¹ <https://less.works/less/framework/overall-retrospective/>

¹² Ken Schwaber, *Agile Project Management with Scrum* (2004)

¹³ Alfie Kohn, *Punished By Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes* (1999)

Conclusioni

Se potete spuntare la maggior parte di questi elementi come raggiunti e avete ancora tempo durante la giornata, fatevi sentire.

Non ci sono formule preconfezionate per creare l'ingegno umano. Questo articolo elenca concetti che potrebbero aiutare, o meno, nella vostra situazione.

Una volta che iniziate a rendervi conto di cosa potreste fare per fare la differenza, potreste scoprire che avete paura di farlo. Questo è un segnale che siete sulla strada giusta.

MODULO SUGLI OSTACOLI ORGANIZZATIVI

PROBLEMA APPARENTE:

RADICE DEL PROBLEMA (USATE CINQUE VOLTE “PERCHÉ?”):

IMPATTO SUL BUSINESS:

IMPATTO EMOTIVO:

RICHIESTA CHIARA, PRATICABILE:

MODULO SUGLI OSTACOLI ORGANIZZATIVI

PROBLEMA APPARENTE:

RADICE DEL PROBLEMA (USATE CINQUE VOLTE “PERCHÉ?”):

IMPATTO SUL BUSINESS:

IMPATTO EMOTIVO:

RICHIESTA CHIARA, PRATICABILE:

MODULO SUGLI OSTACOLI ORGANIZZATIVI

PROBLEMA APPARENTE:

RADICE DEL PROBLEMA (USATE CINQUE VOLTE “PERCHÉ?”):

IMPATTO SUL BUSINESS:

IMPATTO EMOTIVO:

RICHIESTA CHIARA, PRATICABILE:

MODULO SUGLI OSTACOLI ORGANIZZATIVI

PROBLEMA APPARENTE:

RADICE DEL PROBLEMA (USATE CINQUE VOLTE “PERCHÉ?”):

IMPATTO SUL BUSINESS:

IMPATTO EMOTIVO:

RICHIESTA CHIARA, PRATICABILE:

MODULO SUGLI OSTACOLI ORGANIZZATIVI

PROBLEMA APPARENTE:

RADICE DEL PROBLEMA (USATE CINQUE VOLTE “PERCHÉ?”):

IMPATTO SUL BUSINESS:

IMPATTO EMOTIVO:

RICHIESTA CHIARA, PRATICABILE:

MODULO SUGLI OSTACOLI ORGANIZZATIVI

PROBLEMA APPARENTE:

RADICE DEL PROBLEMA (USATE CINQUE VOLTE “PERCHÉ?”):

IMPATTO SUL BUSINESS:

IMPATTO EMOTIVO:

RICHIESTA CHIARA, PRATICABILE:

ISTRUZIONI

Se avete ricevuto questa lista come parte di un corso e il vostro attuale (o ultimo) datore di lavoro ha cercato di fare qualcosa di simile a Scrum, per favore applicate questa lista a quello che avete visto in quel contesto. Contrassegnate ogni elemento come segue:

- √ (per “va bene”)
- Δ (per “potrebbe essere migliorato, e so dove cominciare”)
- ? (per “potrebbe essere migliorato, ma come?”)
- N/A (per “non rilevante” o “non porterebbe alcun beneficio”)

Oppure, se il vostro attuale (o ultimo) datore di lavoro non ha tentato di fare nulla di simile a Scrum, contrassegnate ciascun elemento come segue:

- √ (per “va bene” o “sarebbe facile farlo bene”)
- Δ (per “sarebbe difficile e so da dove cominciare”)
- ? (per “sarebbe difficile e non so da dove cominciare”)
- N/A (per “non rilevante” o “non porterebbe alcun beneficio”)

Quando tutti gli elementi sono stati contrassegnati, riportate da 2 a 6 ostacoli organizzativi sul Modulo sugli ostacoli organizzativi allegato, a prescindere che derivino o meno da questa lista. Scegliete ostacoli per cui avete almeno l'1% di speranza di cambiamento.