



# CHECKLISTA ŚWIĘTNEGO SCRUM MASTERA

# Wstęp

# Moderator na pełny etat?

Przeciętny Scrum Master może pracować z dwoma lub trzema zespołami jednocześnie. Jeżeli odpowiada ci ograniczenie twoich obowiązków do organizowania spotkań, pilnowania ram czasowych i reagowania na przeszkody zgłaszane przez ludzi, to możesz pełnić tę rolę na część etatu. Zespół prawdopodobnie będzie funkcjonował lepiej niż przed wprowadzeniem Scruma i raczej nie wydarzy się żadna katastrofa.

Jeżeli jednak marzy ci się zespół, który w zmieniającej się organizacji osiąga cele dotąd uważane za niewykonalne, to rozważ zostanie świetnym Scrum Masterem.

Świetny Scrum Master pracuje z jednym Zespołem w danej chwili. Szczególnie na początku waszej drogi zalecamy dedykowanego Scrum Mastera dla każdego Zespołu.

Jeśli nie odkryłeś jeszcze jak dużo jest do zrobienia, zwróć uwagę na:

- Product Owenera,
- Zespół,
- praktyki inżynieryjne,
- całą organizację.

## Instrukcja

Zaznacz każdą kratkę jednym z poniższych symboli:

✓ (idzie nam całkiem nieźle)

! (może być lepiej i wiem jak zacząć)

? (może być lepiej, ale nie wiem jak zacząć)

**N/A** (nie dotyczy/nie przyniesie żadnych korzyści)

Na podstawie: Scrum Master Checklist Michaela Jamesa  
(mj4scrum@gmail.com),

<http://scrummasterchecklist.org/>

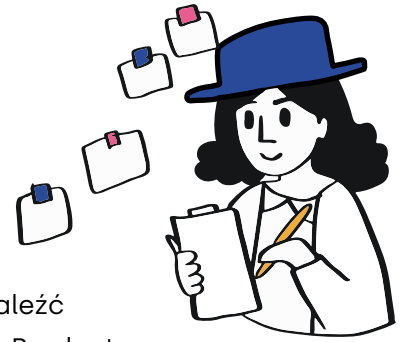
Wersja: Marzec 2023

Na licencji Creative Commons Uznanie autorstwa 3.0 Polska (CC BY 3.0 PL)



## Część I

# Jak sobie radzi Product Owner?



Scrum Master zwiększa efektywność Product Ownera pomagając mu znaleźć sposoby na utrzymanie Backlogu Produktu i Planu Wydań. Pamiętaj, że to Product Owner jest odpowiedzialny za priorytety Elementów Backlogu Produktu.

Czy niezależnie od liczby zespołów, jest jeden (i tylko jeden) Product Owner dla produktu?

Czy Backlog Produktu jest ułożony zgodnie z ostatnią koncepcją Product Ownera?

Czy nowo odkryte wymagania i potrzeby są widoczne w Backlogu Produktu? Pamiętaj: Backlog Produktu nigdy nie jest kompletny i ewoluuje wraz z rozwojem Produktu.

Czy Backlog Produktu jest odpowiednich rozmiarów? Aby osiągnąć zarządzalną liczbę elementów, te na górze Backlogu Produktu powinny być niewielkie i szczegółowe, zaś te na dole – duże i ogólne. Nie ma sensu analizować elementów z dołu Backlogu Produktu, bo plany będą ewoluować w wyniku ciągłej komunikacji między użytkownikami/klientami a ludźmi tworzącymi produkt.

Czy którekolwiek z wymagań (zwłaszcza te z góry Backlogu) może być lepiej przedstawione zgodnie z koncepcją INVEST dla Historii Użytkownika? Czy da się je podzielić na mniejsze, niezależne części, które posiadają wartość dla użytkownika, dają się oszacować i przetestować?

Czy wyjaśniłeś Product Ownerowi co to jest Dług Techniczny i jak go uniknąć? Ważnym elementem może być dołączenie automatycznych testów i refaktoryzacji do Definicji Ukończenia.

Czy Backlog Produktu jest miejscem pozwalającym wszystkim na szybki i łatwy dostęp do informacji?

Czy wszyscy wiedzą jak korzystać z narzędzi do zarządzania Backlogiem Produktu, zwłaszcza jeżeli używasz elektronicznych wersji?

Czy możesz pomóc Product Ownerowi dzielić się istotnymi informacjami, pokazując jak przygotować odpowiednie dane i wykresy?

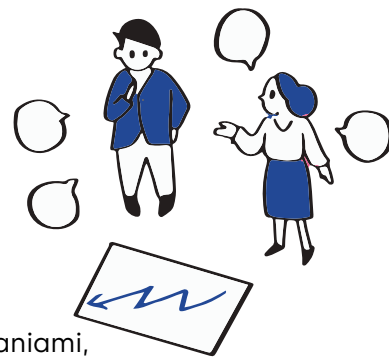
Czy pomagasz Product Ownerowi przyporządkować Elementy Backlogu Produktu do odpowiednich Wydań lub grup priorytetów ?

Czy wszyscy znają faktyczny stan postępu prac? Możesz spróbować pokazać Wykresy Spalania. Wykresy pokazujące ilość faktycznie ukończonych oraz nowo dodanych elementów pozwalają na wczesne wykrycie zmiany zakresu i harmonogramu prac.

Czy Product Owner zaktualizował Plan Wydania po ostatnim Przeglądzie Sprintu? Product Ownerzy, którzy chcą dostarczyć produkt zgodny z oczekiwaniami poświęcają czas na ponowne planowanie Wydania po każdym Sprincie. Wraz z odkrywaniem nowych informacji często pojawia się potrzeba zmiany priorytetów i przesunięcia części pracy do następnych wydań.

## Część II

# Jak sobie radzi Zespół?



Możesz dać dobry przykład współpracując z członkami Zespołu nad ich zadaniami, istnieje jednak ryzyko, że za bardzo skupisz się na tematach technicznych. Dlatego weź też pod uwagę inne obowiązki w stosunku do Zespołu.

Czy członkowie Zespołu lubią się wzajemnie, spędzają wspólnie czas i świętują sukcesy?

Czy członkowie Zespołu wzajemnie pilnują utrzymywania wysokich standardów? Czy stawiają przed sobą wyzwania prowadzące do rozwijania umiejętności?

Czy są problemy lub możliwości, których omawiania Zespół unika, ponieważ powodują zbyt duży dyskomfort?

Czy próbowałeś różnych sposobów na przeprowadzenie Retrospektywy?

Czy Zespół skupia się na Celu Sprintu?

Czy Tablica Zadań Sprintu odzwierciedla to, co zespół faktycznie robi? Wystrzegaj się ukrytych zadań i takich, które zajmują więcej niż jeden dzień pracy. Zidentyfikuj zadania nie związane z zobowiązaniami Zespołu na obecny Sprint. Stanowią one przeszkodę dla realizacji Celu Sprintu.

Czy Zespół składa się z 3-9 osób. Czy posiada wszystkie umiejętności niezbędne do zbudowania Przyrostu Produktu przygotowanego do dostarczenia po każdym Sprincie?

Czy plan Sprintu jest aktualny, a Tablica Zadań pokazuje bieżącą sytuację?

Czy narzędzia ułatwiające Zespołowi samodzielną pracę (Tablica Zadań, Wykres Spalania Sprintu, Lista Przeszkód, itd.) są widoczne, dostępne dla członków Zespołu i wygodne w użyciu?

Czy te artefakty są odpowiednio chronione przed osobami z zewnątrz? Nadmierna kontrola przez osoby spoza Zespołu może utrudniać wewnętrzną przejrzystość i samodzielną pracę.

Czy członkowie Zespołu sami wybierają zadania do zrobienia?

Czy potrzeba spłaty Długu Technicznego została ujęta w Definicji Ukończenia? Czy kod stopniowo staje się bardziej czytelny i łatwiejszy w utrzymaniu?

Czy członkowie Zespołu zostawiają swoje tytuły i stanowiska za drzwiami, wspólnie odpowiadając za całość pracy (testowanie, prowadzenie dokumentacji, itd.)?

## Część III

# Jak wyglądają praktyki inżynierskie?



Czy środowisko deweloperskie daje każdemu możliwość łatwego, automatycznego sprawdzenia czy zmiany nie uszkodziły wcześniej działającej funkcjonalności (testy regresji)? Zwykle jest to osiągnięte poprzez wykorzystanie środowisk xUnit (JUnit, NUnit, itp.).

Czy utrzymujecie odpowiednią równowagę pomiędzy automatycznymi testami funkcjonalnymi (end-to-end) i testami jednostkowymi?

Czy cały Zespół zna język, w którym pisane są testy funkcjonalne i jednostkowe? Autorskie języki skryptowe i narzędzia do nagrywania testów obsługiwane przez nielicznych członków zespołu utrudniają współpracę. Rozwiązaniem może być tworzenie testów w tym samym języku, w którym jest tworzony produkt.

Czy Zespół odkrył użyteczną przestrzeń pomiędzy testami systemowymi a testami jednostkowymi?

Czy system Ciągłej Integracji automatycznie powiadamia, gdy ktoś spowoduje błąd regresji? Czy ta pętla informacji zwrotnej może zostać jeszcze bardziej skrócona? („Codzienne kompilacje są dla mięczaków” – Kent Beck).

Czy wszystkie testy są częścią systemu Ciągłej Integracji?

Czy członkowie Zespołu doceniają wartość ciągłego projektowania i refaktoryzacji, jako alternatywy do projektowania wszystkiego z góry? Pełne pokrycie automatycznymi testami pozwala na bezpieczną refaktoryzację. Natomiast zaniedbanie refaktoryzacji utrudnia zmianę produktu w przyszłości, zwłaszcza że ciężko znaleźć dobrych programistów chcących pracować nad złym kodem.

Czy Definicja Ukończenia dla każdego Elementu Backlogu Produktu zawiera pokrycie kodu automatycznymi testami i refaktoryzacją? Korzystanie z Test Driven Development (TDD) zwiększa prawdopodobieństwo sukcesu.

Czy członkowie Zespołu pracują w parach lub w grupie (Pair Programming, Mob Programming) przez większość czasu? Używanie tej metody może znacząco ułatwić utrzymanie kodu oraz zmniejszyć ilość błędów. Ponieważ stawia to ludziom nowe wyzwania na początku jest niekomfortowe i może sprawiać wrażenie, że zabiera więcej czasu (tylko jeżeli mierzymy efektywność ilością linii kodu, a nie ukończonymi funkcjami). Daj dobry przykład programując w parze z kolejnymi osobami z Zespołu. Niektórzy z nich zaczną korzystać z tej techniki na stałe.

## Część IV

# Jak radzi sobie organizacja?



Czy Zespoły komunikują się między sobą w wystarczający sposób? „Scrum of Scrums” jest tylko jednym ze sposobów na wymianę informacji i niekoniecznie najlepszym.

Czy Zespoły mogą niezależnie od siebie dostarczać działające funkcje, w miarę potrzeby modyfikując różne elementy architektury?

Czy przeprowadzacie Wspólną Retrospektywę (Overall Retro) w celu rozwiązania międzyzespołowych i systemowych problemów w organizacji?

Jeżeli jest taka potrzeba, czy przeszkody organizacyjne są umieszczone na ścianie gabinetu dyrektora działu/firmy? Czy ich koszt może być przedstawiony w gotówce, utraconym czasie lub jakości, albo utraconych szansach na pozyskanie klienta. Jednak ucz się na błędach Kena Schwabera – „Martwy Scrum Master to bezużyteczny Scrum Master”.

Czy organizacja oferuje ścieżki kariery zgodne z celami Zespołu? Jeżeli promuje ścieżki związane z programowaniem i architekturą kosztem testów, automatyzacji środowisk lub dokumentacji dla użytkownika, to odpowiedź brzmi „nie”.

Czy organizacja została uznana przez prasę branżową lub inne niezależne źródło jako jedno z najlepszych miejsc pracy lub lidera w swojej branży?

Czy tworzycie uczącą się organizację?



**Więcej darmowych  
materiałów znajdziesz na:**

