

Programowanie zespołowe

Laboratorium 6 - scrum cz. 2 i przegląd wybranych metodyk
zwinnych

mgr inż. Krzysztof Szwarc

krzysztof@szwarc.net.pl

Sosnowiec, 28 marca 2017

Zdarzenia w scrumie

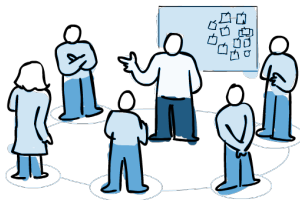
Zdarzenia mają za zadanie wprowadzenie regularności oraz ograniczenia potrzeby organizowania innych spotkań. Są one ograniczone czasowo (ang. timebox), a maksymalny czas ich trwania jest ustalany z góry.

- Czas trwania sprintu jest ustalany w chwili jego rozpoczęcia i nie może być modyfikowany.
- Pozostałe zdarzenia mogą się zakończyć w momencie, gdy cel zostanie osiągnięty (wykorzystanie wyłącznie niezbędnej ilości czasu).



Rodzaje zdarzeń

- 1 Sprint (zawiera wszystkie zdarzenia, role i artefakty).
- 2 Planowanie sprintu.
- 3 Codzienny Scrum.
- 4 Przegląd sprintu.
- 5 Retrospektywa sprintu.



<http://csharpcorner.mindcrackerinc.netdna-cdn.com>

Cechy przebiegu (ang. sprint)

- Trwa maksymalnie jeden miesiąc kalendarzowy.
- Najczęściej trwa od jednego do czterech tygodni.
- Zaleca się stosowanie przebiegów o stałych długościach.

Planowanie sprintu

- Określenie jaki będzie rezultat następnego sprintu (zdefiniowanie następnego Przyrostu Produktu).
- Stworzenie Rejestru Sprintu przez zespół, który wybiera zadania z Rejestru Produktowego.
- Ograniczony najczęściej do ośmiu godzin dla miesięcznego sprintu (dla krótszych przebiegów jest ono zwykle krótsze).
- Rolą Scrum Mastera jest zapewnienie, że Planowanie sprintu się odbywa i jego uczestnicy rozumieją cel tego zdarzenia.
- Scrum Master uczy Zespół Scrumowy utrzymywać je w wyznaczonych ramach czasowych.

Codzienny Scrum

- Przeznaczony dla Zespołu Deweloperskiego.
- Ograniczony czasowo do 15 minut.
- Ma wyznaczone stałe miejsce i czas.
- Podczas niego następuje synchronizacja działań i stworzenie planu na najbliższe 24 godziny (dokonywana jest inspekcja prac wykonanych od ostatniego Codziennego Scruma i prognozowanie prac, które mogą zostać wykonane przed kolejnym spotkaniem).
- Scrum Master zapewnia, że Zespół Deweloperski (i tylko on) spotyka się w ramach Codziennego Scruma, lecz to Zespół Deweloperski jest odpowiedzialny za jego przebieg. Scrum Master uczy Zespół Deweloperski także jak utrzymać piętnastominutowe ograniczenie czasowe.
- Kluczowy dla procesu inspekcji i adaptacji.

Codzienny Scrum - trzy pytania

Każdy członek Zespołu Deweloperskiego udziela odpowiedzi na trzy pytania:

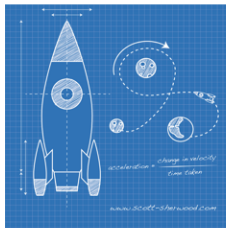
- 1 Co zrobiłem wczoraj, co pomogło Zespołowi Deweloperskiemu przybliżyć się do osiągnięcia celu sprintu?
- 2 Co zrobię dzisiaj, co pomoże Zespołowi Deweloperskiemu przybliżyć się do osiągnięcia celu sprintu?
- 3 Czy dostrzegam przeszkody mogące uniemożliwić osiągnięcie celu sprintu?



- Organizowany na zakończenie sprintu w celu przeprowadzenia inspekcji Przyrostu Produktu i ewentualnego dostosowania Rejestru Produktowego.
- Zespół Scrumowy i interesariusze dyskutują o efekcie sprintu (uzyskanie informacji zwrotnej).
- Uczestnicy spotkania wspólnie decydują co należałoby wykonać w następnej kolejności, aby zwiększyć dostarczaną wartość.
- Trwa maksymalnie 4 godziny dla miesięcznego sprintu (krócej dla krótszych sprintów).
- W jego rezultacie aktualizowany jest Rejestr Produktowy.

Przegląd sprintu cz. 2

- Scrum Master dba o to, aby spotkanie się odbyło i jego uczestnicy rozumieli jego cel. Uczy on wszystkich zgromadzonych jak utrzymywać je w wyznaczonych ramach czasowych.



<http://www.full-stackagile.com>

Przegląd sprintu - przebieg

- 1 Właściciel Produktu informuje o stworzonej funkcjonalności.
- 2 Zespół Deweloperski przedstawia co poszło dobrze, a co źle podczas sprintu.
- 3 Zespół Deweloperski przedstawia wynik pracy i odpowiada na pytania w sprawie Przyrostu Produktu.
- 4 Właściciel Produktu omawia Rejestr Produktowy i, jeśli jest to wymagane, przewiduje termin zakończenia prac.
- 5 Cała grupa wspólnie omawia kolejne kroki procesu.
- 6 Dokonuje się oceny zmian rynku i rewiduje zasoby.

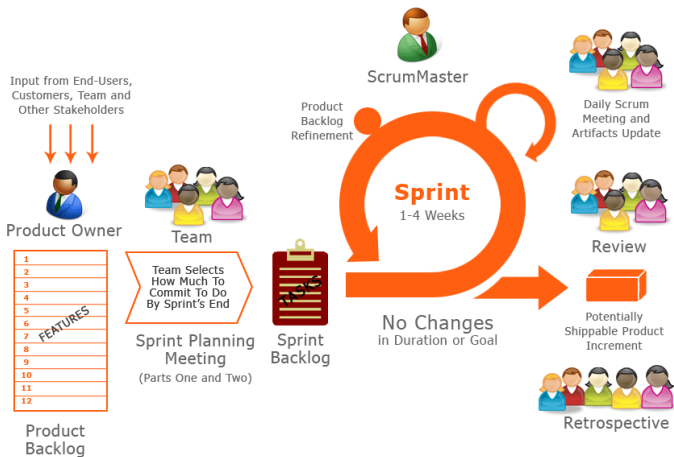
Retrospektywa sprintu

- Ma na celu przeprowadzenie inspekcji działań i opracowania planu usprawnień.
- Trwa maksymalnie 3 godziny dla comiesięcznych sprintów.
- Scrum Master zapewnia, że Retrospektywa się odbędzie, a uczestnicy rozumieją jej cel i uczy jak utrzymać zdarzenie w określonych ramach czasowych. Uczestniczy on w Retrospektywie jako zwykły członek zespołu.



<http://4.bp.blogspot.com/>

Podsumowanie - scrum

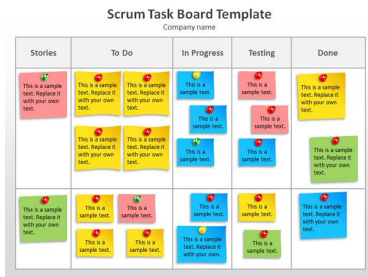


<http://www.agilebuddha.com/>

Dodatkowe narzędzie - tablica scrum

Tablica scrum

Tablica scrum (ang. scrum board) wykorzystywana jest do zobrazowania Rejestru Sprintu w formie listy zadań z podziałem najczęściej na aktywności do wykonania, wykonywane oraz zakończone. Opcjonalnie można na niej zwizualizować np. tematy do omówienia podczas Retrospekcji.



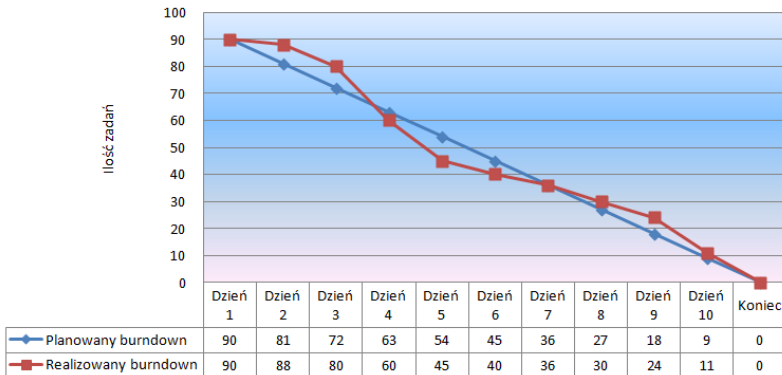
Wykres wypalenia

Wykres wypalenia (ang. burndown chart) to metoda wizualizacji postępów sprintu. W zależności od podejścia możemy korzystać z wykresów opartych na zadaniach lub na Story Points. Cechy wykresu:

- Przedstawiając odchylenia względem idealnej linii postępu możemy wyciągać wnioski podczas Retrospektywy co do ważnych momentów sprintu.
- Ukazuje ryzyko niemożności wykonania prac w założonym terminie.
- Prezentuje status sprintu.

Przykładowy wykres wypalenia

Burndown chart dla zadań w iteracji 2 tygodniowej



<http://3.bp.blogspot.com>

- 1 Przeprowadźcie - w swoich zespołach - pierwsze Planowanie sprintu. Dla zbyt dużych historyjek użytkownika dokonajcie ich dekompozycji, zmieniając je na Epici.

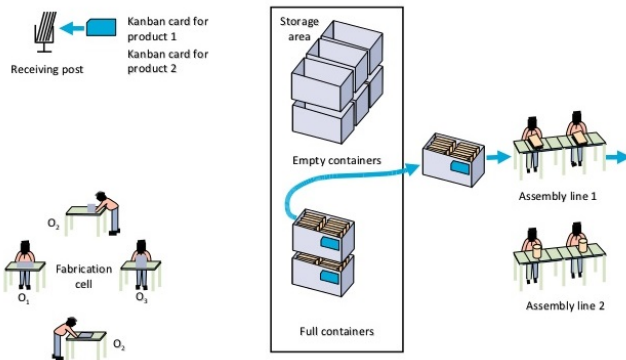
- 1 Przeprowadźcie - w swoich zespołach - pierwsze Planowanie sprintu. Dla zbyt dużych historyjek użytkownika dokonajcie ich dekompozycji, zmieniając je na Epici.
- 2 Przedstaw wyniki zadania 1 na tablicy scrumowej. Skorzystaj w tym celu z <https://trello.com/>

Kanban w logistyce produkcji

Kanban jest metodą sterowania produkcją, która została opracowana w latach pięćdziesiątych w Japonii i opiera się na kartach wyrobów oraz ich cyrkulacji i analizie. Jego idea opiera się na hasle „7 razy żadnych”:

- Żadnych braków.
- Żadnych opóźnień.
- Żadnych zapasów.
- Żadnych kolejek.
- Żadnych beczynności.
- Żadnych zbędnych operacji technologicznych i kontrolnych.
- Żadnych przemieszczeń.

The Single-Card Kanban System



<https://www.slideshare.net/>

Kanban w inżynierii oprogramowania

Kanban pozwala na utrzymanie pełnej kontroli nad procesem tworzenia oprogramowania, umożliwia eliminację przyczyn nieefektywności i zwiększenie produktywności. Opiera się na trzech zasadach:

- Wizualizacja (przedstawienie kolejnych etapów procesów na tablicy).
- Ograniczenie pracy w toku (ustalenie maksymalnej liczby zadań, które mogą znajdować się w danej kolumnie).
- Zarządzanie strumieniem (systematyczny pomiar czasu i płynności wykonywania zadań, w celu optymalizacji procesów).

Tablica kanbanowa



<http://procognita.pl>

Kanban, a scrum

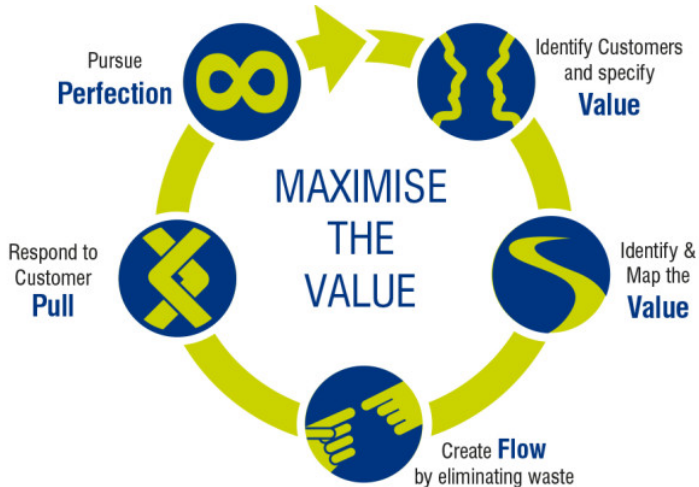
- Kanban jest o wiele mniej sformalizowany (brak estymacji, sprintów itp.).
- Kanban jest bardziej elastyczny (brak sprintów, członkowie zespołu mogą brać udział w innych projektach).
- Scrum lepiej nadaje się do większych projektów.

Lean management

Szczupłe zarządzanie (ang. Lean management) jest koncepcją zarządzania przedsiębiorstwem, zakładającą dostosowanie go do warunków rynku na drodze przekształceń organizacyjnych i funkcjonalnych. Zakłada ona:

- Elastyczność struktury organizacyjnej.
- Ciągłe doskonalanie organizacji za pomocą szkoleń kadry i pracowników.
- Wyodrębnienie małych jednostek organizacyjnych oraz zespołów pracujących nad określonym zadaniem.
- Podział odpowiedzialności.

Zasady lean management



<http://www.madabout.org.uk/>

Szczupłe programowanie (ang. Lean software development) powstało poprzez zastosowanie lean management w procesie produkcji oprogramowania. Opiera się ono na 7 zasadach:

- Eliminacja strat.
- Tworzenie jakości i spójności (równowaga między funkcjonalnością, niezawodnością i wartością).
- Wzmocnienie pozyskiwania wiedzy (uzyskiwanie informacji zwrotnej od klienta).
- Podejmowanie decyzji najpóźniej, jak to możliwe.
- Wdrażanie najwcześniej, jak to możliwe.
- Respektowanie zespołu (wspieranie zespołu, pozwolenie na samoorganizację).
- Spojrzenie na całość (kompleksowe ujęcie).

THINK LESS IS MORE – ELIMINATE WASTE

The 7 Wastes of Manufacturing

- Inventory
- Extra Processing
- Overproduction
- Transportation
- Waiting
- Motion
- Defects

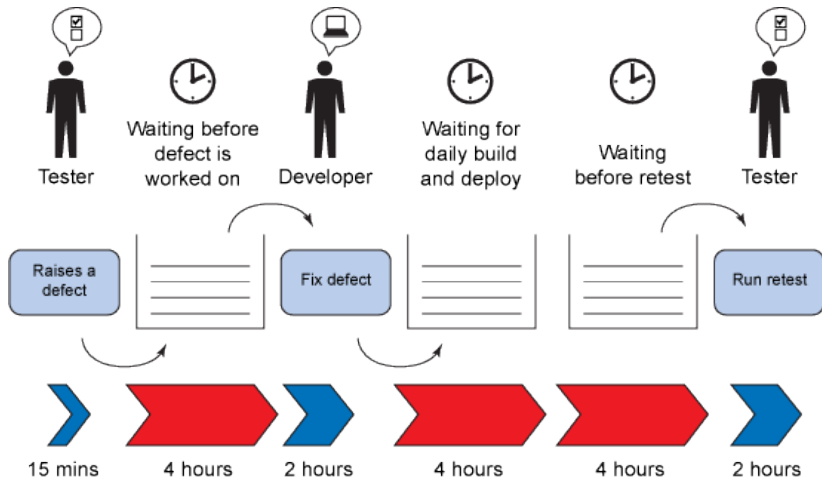
The 7 Wastes of Software Development

- Partially Done Work
- Extra processes
- Extra features
- Task Switching
- Waiting
- Motion
- Defects



<https://bgoerke.files.wordpress.com/>

Marnotrawstwo (Muda) - przykład



<https://www.ibm.com/>

Dziękuję za uwagę