

Zaawansowana inżynieria oprogramowania

Wykład

SCRUM – zwinna metoda tworzenia oprogramowania

Marek Krętowski
pokój 206
e-mail: m.kretowski@pb.edu.pl
http://aragorn.pb.bialystok.pl/~mkret

Wersja 0.9

Wprowadzenie

- Metodyka adaptacyjna stosowana w procesie wytwarzania złożonych produktów
 - Daje konkretny i spójny przepis na prowadzenie projektów inf.
 - Pozostawia dużą swobodę dotyczącą sposobu realizacji w konkretnej organizacji
- Wg twórców opisuje ramy metodyczne, w ramach których możliwe jest stosowanie różnych procesów i technik
 - Rola Scrum-a jest wykazywanie nieefektywności stosowanych praktyk tak, aby możliwe było ich usprawnienie
- Korzyści z dobrej implementacji Scrum-a:
 - zwiększenie wydajności zespołów i duża elastyczność przy realizacji
- Dotyczy **wyłącznie fazy wytwarzania**
- Nie dotyczy przygotowania wizji projektu, podejmowania decyzji strategicznych czy zarządzania kontraktem

Zaawansowana inżynieria oprogramowania

2 / 17

Historia

- Korzenie: w połowie lat 80-tych Japończycy (Takeuchi i Nonaka) opublikowali artykuł w *Harvard Business Review* opisujący nowy sposób realizacji projektów:
 - przez mały interdyscyplinarny zespół w ramach kolejnych zachodzących na siebie faz
 - odniesienie do gry w rugby
- Narodziny: **1995 r.**
 - pierwszy raz zaprezentowana publicznie kompletna wizja metodyki na konferencji OOPSLA w Austin
- Ojcowie:
 - Jeff **Sutherland**
 - Ken **Schwaber** (aktualnie prezydent organizacji Scrum Alliance)
- Pierwsi odważni: Individual Inc., Fidelity Investment oraz IDX (GE Medical)
- Obecnie używany nie tylko w małych organizacjach, ale też w wielkich korporacjach: np. Fujixerox, Canon, Honda, HP, IBM i wielu innych

Zaawansowana inżynieria oprogramowania

3 / 17

Trochę ideologii, czyli 3 filary Scrum

- Scrum bazuje na teorii empirycznej kontroli procesu
 - wykorzystuje iteracyjność, aby osiągnąć lepszą przewidywalność i kontrolę ryzyka
- Reguła **przejrzystości (1)**:
 - wszystkie te aspekty procesu, które wpływają na jego wynik są widoczne dla zarządzających rezultatami procesu
 - musi być jasne co jest konkretnym wynikiem, rezultatem procesu (zgodność z definicją wykonalności zarządzających; *DoD*)
- **Kontrola (2)** – wiele aspektów procesu musi być sprawdzanych (inspekcje), aby zapewnić wykrywanie niedopuszczalnych odchyień:
 - samo kontrolowanie wpływa na proces => odpowiednia częstotliwość, umiejętności oraz staranność kontrolujących
- **Adaptacja (3)**
 - stwierdzenie, że aspekt procesu nie spełni kryterium dopuszczalności => produkt nieakceptowalny wymaga dostosowania procesu lub materiału wejściowego
 - dostosowanie powinno być szybkie => minimalizacja dalszych odchyień

Zaawansowana inżynieria oprogramowania

4 / 17

Trzy role: Właściciel produktu, Mistrz i Zespół

- **Mistrz** (Scrum Master) – zadanie: dopilnowanie przestrzegania przez zespół wartości praktyk i reguł Scrum-a
 - uczy Zespół, trenując i prowadząc członków do osiągnięcia wyższej wydajności i tworzenia produktów o wyższej jakości
 - usuwanie przeszkód w środowisku, które nie jest przystosowane do wytwarzania złożonych produktów; przywództwo służebne
 - współpracując z klientami i kierownictwem identyfikuje osobę, która będzie Właścicielem Produktu; następnie uczy ją wykonywania pracy w ramach Scrum (odpowiada za nauczanie)
- Samoorganizujący się i interdyscyplinarny (międzyfunkcyjny) **Zespół**

Zaawansowana inżynieria oprogramowania

5 / 17

Właściciel produktu

- **Właściciel produktu** (Product Owner) – zarządza rejestrem produktowym (ma być widoczny i zrozumiały) i czuwa na wartością pracy wykonywanej przez zespół
 - jest odpowiedzialny za reprezentowanie interesów wszystkich zainteresowanych projektem i tworzonym systemem
 - jedna osoba, nie komitet; mogą istnieć osoby doradzające, ale aby zmienić priorytet pozycji rejestru decyzję musi podjąć Właściciel
 - aby odniósł sukces jego decyzje muszą być respektowane; nie może być innych decydentów, którzy ustawiają pracę zespołu

Zaawansowana inżynieria oprogramowania

6 / 17

Członkowie zespołu

- Zespół programistów przetwarza podczas Sprintów rejestr produktowy w kolejne przyrosty funkcjonalności
 - członkowie muszą posiadać wszystkie umiejętności do wytworzenia kolejnego przyrostu => mogą mieć wyspecjalizowane umiejętności,
- Ważne umiejętności wspólne dla wszystkich
 - Zespołowe podjęcie zobowiązania i przekształcenie wymagań w używalny produkt
 - osoby, które nie chcą kodować, a tylko np. projektować nie pasują do zespołu;
 - każdy musi dołożyć cząstkę pracy, nawet kosztem nauki lub przypomnienia
- Nie istnieje hierarchia służbowa, nie ma stanowisk, nie ma podziału funkcjonalnego
- Zespół samoorganizujący się – każdy członek wykorzystuje swoje kompetencje do rozwiązania problemu; powstająca synergia podnosi wydajność
- Optymalna liczebność zespołu: 7 +/- 2 (bez Mistrza i Właściciela):
 - Mniej niż 5: mniej interakcji => mniejsza produktywność; możliwe niedobory umiejętności
 - Więcej niż 9: wymagana intensywniejsza koordynacja

Zaawansowana inżynieria oprogramowania

7/17

Bajka o kurczaku i świnkach



[A. Koszljajda „Zarządzanie projektami IT”]

- Należy jasno rozgraniczyć osoby, które rzeczywiście pracują (poświęcają się) przy realizacji projektu (członkowie zespołu - „świnki”) od osób, które są zaangażowane, ale nie pracują bezpośrednio nad projektem („kurczaki”)
- Interesanci – odpowiedzialni za podejmowanie decyzji strategicznych
 - za ich taktyczną implementacją odpowiedzialny jest właściciel produktu
 - mogą biernie obserwować codzienne spotkania

Zaawansowana inżynieria oprogramowania

8/17

Podstawowe narzędzia

- Rejestr produktowy**, dziennik zadań produktu, zaległości produktowe (Product Backlog) – zhierarchizowana lista wszelkich elementów składających się na ostateczną postać produktu
 - definiuje wysokopoziomowe, biznesowe funkcje aplikacji, które mają zostać wykonane w trakcie całego projektu
 - podstawowa forma dyskusyjna pomiędzy zespołem a właścicielem produktu
 - żywy dokument – zmiany w wymaganiach
- Rejestr zadaniowy**, dziennik zadań Sprinta (Sprint Backlog) – lista zadań (technicznych), których wykonanie doprowadzi do przetworzenia wybranych (w danym Sprincie) pozycji rejestru produktowego w gotowy fragment potencjalnie zbywalnego produktu
 - optymalna długość trwania pojedynczego zadania od 4 do 12 godzin przy 14 dniowym Sprincie

Zaawansowana inżynieria oprogramowania

9/17

Sprint i rejestr zadaniowy

Task Description	Originator	Responsible	Status (Not Started / In Progress / Completed)	Hours of work remaining													
				1	2	3	4	5	6	7	8	9	10	11	12		
Need to discuss the goals and features for Sprint 2.0	John	Alisa	Not Started	8	8	8	8	8	8	8	8	8	8	8	8	8	
Get KES Data	John	Alisa	Completed	12	0	0	0	0	0	0	0	0	0	0	0	0	
Analyze KES Data - Time	George	George	In Progress	24	24	24	24	12	0	0	0	0	0	0	0	0	
Analyze KES Data - Revenue	John	John	In Progress	12	12	12	12	12	12	12	12	12	12	12	12	12	
Analyze KES Data - Conversion	Alisa	Alisa	In Progress	24	24	24	24	12	0	0	0	0	0	0	0	0	
Analyze KES Data - Profitability	Alisa	Alisa	In Progress	24	24	24	24	12	0	0	0	0	0	0	0	0	
Define a test database	Barry/John	Barry/John	In Progress	80	80	80	80	40	40	40	40	40	40	40	40	40	
Utilize the use of the KES database	John	John	Not Started	8	8	8	8	8	8	8	8	8	8	8	8	8	
Look at KES Data via the UI	John	John	In Progress	3	3	3	3	3	3	3	3	3	3	3	3	3	
Confirm agreement with KES	John	John	Not Started	8	8	8	8	8	8	8	8	8	8	8	8	8	
Confirm KES Data availability	John	John	Not Started	1	1	1	1	1	1	1	1	1	1	1	1	1	
Switch DB to 3.1. Run all tests	Alisa	Alisa	Not Started	8	8	8	8	8	8	8	8	8	8	8	8	8	
Store PDF files in a structure	Barry/John	Barry/John	Completed	8	0	0	0	0	0	0	0	0	0	0	0	0	
Display Content grid on all webpages/pages	Barry/John	Barry/John	Completed	4	0	0	0	0	0	0	0	0	0	0	0	0	
Make test data reusable	Barry	Barry	In Progress	10	10	10	10	10	10	10	10	10	10	10	10	10	
Allow registration and database to 'load content'	Richard	Richard	Completed	4	4	4	4	4	4	4	4	4	4	4	4	4	
Set up Content environment	John	John	Completed	2	2	2	2	2	2	2	2	2	2	2	2	2	
Test sign in flow	John	John	In Progress	25	25	25	25	25	25	25	25	25	25	25	25	25	
Define sign up flow required for content in 2012	Lynne	Lynne	In Progress	40	40	40	40	40	40	40	40	40	40	40	40	40	
Release content for report generation	John	John	In Progress	12	12	12	12	12	12	12	12	12	12	12	12	12	
Build standard report exception generation	John	John	In Progress	12	12	12	12	12	12	12	12	12	12	12	12	12	
Health evaluate the results on report grid	Alisa	Alisa	Completed	4	4	4	4	4	4	4	4	4	4	4	4	4	
Modify Content/Content to AWS 6.0 (AWS - T1) server	Alisa	Alisa	Not Started	4	4	4	4	4	4	4	4	4	4	4	4	4	
Create web server for Content in AWS	Alisa	Alisa	Completed	1	0	0	0	0	0	0	0	0	0	0	0	0	
LTCS Data	George	George	In Progress	12	12	12	12	12	12	12	12	12	12	12	12	12	
Follow up with questions about KES data in test form on AWS T2	Alisa	Alisa	Completed	10	10	10	10	10	10	10	10	10	10	10	10	10	
Map KES data to Action Tables - new	Alisa	Alisa	In Progress	10	10	10	10	10	10	10	10	10	10	10	10	10	
Prepare SQL to import from KES tables to Action Tables	George	George	In Progress	25	25	25	25	25	25	25	25	25	25	25	25	25	

Rejestr zadaniowy

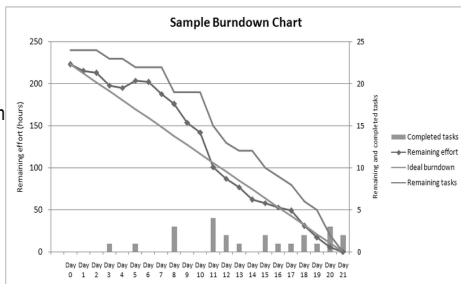
[K. Schwaber „Sprawne zarządzanie ...”]

Zaawansowana inżynieria oprogramowania

10/17

Podstawowe narzędzia (2)

- Wykres wypalania**, diagram spalania (Burndown Chart) – ocena postępu prac w projekcie
 - Pomiar wypalania dla projektu (Release Burndown) – pokazuje elementy pozostające w rejestrze produktowym w stosunku do czasu zakończenia projektu
 - Pomiar wypalania dla Sprintu (Sprint Burndown) – pokazuje zadania pozostające w rejestrze zadaniowym w stosunku do końca Sprintu



[Wikipedia „Burn down chart”]

Zaawansowana inżynieria oprogramowania

11/17

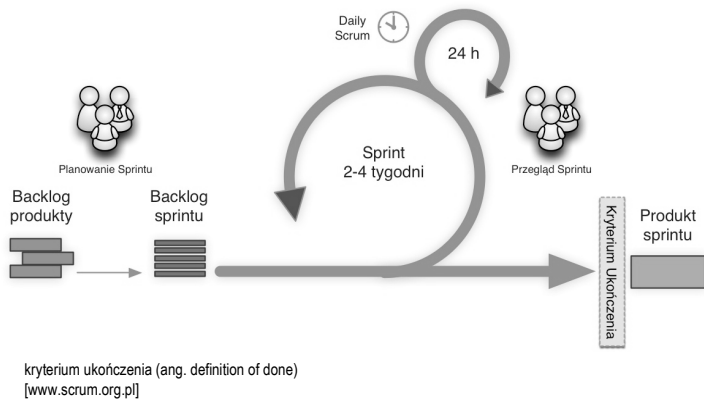
Spotkanie planistyczne Sprintu

- Gdy trzeba przygotować nową iterację; czas 8 godzin (2 x 4)
- Część druga (sposób realizacji)
 - Identyfikowane zostają zadania (szczegółowe opisy prac); krótki czas realizacji
 - Właściciel objaśnia rejestr produktowy; kompromis pomiędzy jego oczekiwaniami a możliwościami produkcyjnymi Zespołu
 - Zespół organizuje się sam, aby podjąć się realizacji
 - Możliwość negocjacji z Właścicielem, gdy założony zakres nieodpowiedni
- Część pierwsza (planowanie):
 - właściciel przedstawia pozycje rejestru produktowego o najwyższym priorytecie
 - zespół i właściciel współpracują by ustalić, która funkcjonalność na być stworzona (ale decyzja właściciela)
 - wyбір Celu sprintu (Sprint Goal) – ma uzmysławiać i przypominać zespołowi po co buduje dany przyrost produktu

Zaawansowana inżynieria oprogramowania

12/17

Mechanika Sprintu



kryterium ukończenia (ang. definition of done)
[www.scrum.org.pl]

Zaawansowana inżynieria oprogramowania

13 / 17

Codzienny Scrum

- Krótkie (15 minut) spotkanie (inspekcyjno-adaptacyjne)
 - zwykle o stałej porze, na stojąco
 - tylko członkowie zespołu
- Każdy z członków odpowiada na 3 pytania:
 - Co zrobił od ostatniego spotkania?
 - Co będzie robił do następnego spotkania?
 - Jakie są przeszkody, zagrożenia w realizacji planów?
- Okazja do kontroli postępu prac ku przyjętemu celowi Sprintu:
 - poprawia komunikację, eliminuje potrzebę innych spotkań, identyfikuje i usuwa przeszkody, wspiera szybkie decyzje, podnosi poziom znajomości stanu prac projektowych



- Mistrz odpowiada za doprowadzenie do spotkania i je nadzoruje organizacyjnie
 - zwięzłość wypowiedzi, egzekwowanie reguł

Zaawansowana inżynieria oprogramowania

14 / 17

Przeгляд Sprintu

- Sprint Review
- Pod koniec Sprintu; ograniczenie czasowe np. 4 g.; spotkanie nieformalne
- Zespół oraz interesariusze podsumowują wykonaną pracę
- Na podstawie aktualnego stanu i zmian w rejestrze produktowym ustalają co ma być wykonane w kolejnych Sprintach
- Właściciel produktu stwierdza, które funkcjonalności zostały (nie) wykonane
- Zespół omawia prace zakończone sukcesem oraz napotkane problemy i metody ich rozwiązania
- Właściciel omawia stan rejestru produktowego
 - Wysuwa propozycje prawdopodobnych terminów zakończenia (przy różnych założeniach dotyczących tempa)
- Cała grupa omawia fakty i ich istotność w ustaleniu zakresu kolejnych prac

Zaawansowana inżynieria oprogramowania

15 / 17

Retrospektywa Sprintu

- Po przeглядzie, ale przed kolejnym spotkaniem planistycznym
 - ograniczenie czasowe – 3 g.
 - Mistrz zachęca członków do przejrzenia pracy programistycznej, aby w kolejnym Sprintcie być bardziej efektywnym
- Celem jest sprawdzenie przebiegu ostatniego Sprintu pod kątem osób biorących udział w przedsięwzięciu, ich relacji, procesu i narzędzi
- Identyfikacja i hierarchizacja elementów pozytywnych i takich, które można by zrealizować lepiej:
 - skład zespołu, rozkład spotkań, narzędzia, definicji gotowości, metod komunikacji i procesów
- Ustalenie kroków naprawczych do stosowania w kolejnych Sprintach

Zaawansowana inżynieria oprogramowania

16 / 17

Podsumowanie

- Cecha charakterystyczną Scrum jest duży stopień nieformalności i koncentracja na efektach
- Może to oznaczać wysoką produktywność, ale także dużą odpowiedzialność
 - kluczowa jest wysoka motywacja i bieżąca komunikacja z klientem
- W odniesieniu do XP jest rozwiązaniem komplementarnym
 - Koncentruje się na procesie a nie na dobrych praktykach
 - Często organizacje implementują Scrum i wybrane praktyki XP

Zaawansowana inżynieria oprogramowania

17 / 17

Przygotowano na podstawie:

- *Zarządzanie projektami IT. Przewodnik po metodykach*, A. Koszłajda, Helion, 2010.
- *SCRUM guide – Przewodnik po metodyce SCRUM*, K. Schwaber, J. Sutherland; tłum. A. Gajewska, T. Włodarek, 2010.
- *Sprawne zarządzanie projektami metodą Scrum*, K. Schwaber, Microsoft Press, 2005.