Daniel Górski
Korporacyjna Java
Wykład 2: Metodologie

Jak przechowywać kod

- Pliki
- RCS(Revision Control System)
- CVS(Concurrent Versions System)
- Gerrit
- SVN(Subversion)
- GIT

Model kaskadowy (Waterfall model)

- Planowanie systemu (Zaczynając od specyfikacji wymagań)
- Analiza systemu (Czy można to zrobić)
- Projekt systemu
- Implementacja
- Testowanie (Integracja)
- Wdrożenie i utrzymanie

PRINCE2

- PRojects IN Controlled Environments
- Ogólna metodyka zarządzania projektami biznesowymi
- Określony czas trwania
- Zdefiniowane i mierzalne wyniki
- Pula zasobów
- Struktura organizacyjna

Metodologie zwinne

- Dlaczego? Perspektywa programisty:
  - Bardzo trudne jest zaprojektowanie dużego systemu w całości
    - Zwłaszcza bez szczegółowych wymagań
  - Wiele niewiadomych na etapie projektowania
    - Szacowanie potrzebnego czasu jest trudne / obarczone dużym ryzykiem
    - Branie odpowiedzialności za coś nierozpoznanego
  - Pisanie pilotów znacząco zmniejsza późniejsze ryzyko, ale samo w sobie jest kosztowne

Metodologie zwinne

- Dlaczego? Perspektywa biznesu:
  - Budżet i czas projektów tworzonych w sposób kaskodowy regularnie się rozjeżdża
  - Specyfikacja wymagań biznesowych dla dużego projektu nie jest łatwą sprawą
    - W detalach biznes nie wie / nie ma wyobrażenia co dokładnie jest potrzebne
    - Osoba dająca wymagania nie musi być "użytkownikiem końcowym"
  - Startup zwykle poszukuje, a nie wie dokładnie "czego chce" / "czego chcą jego klienci"

Manifest Agile

- Ludzie i interakcje ponad procesy i narzędzia
- Działające oprogramowanie ponad szczegółową dokumentację
- Współpracę z klientem ponad negocjacje umów
- Reagowanie na zmiany ponad realizację założonego planu
  - To jest kluczowe z perspektywy programisty: reagowanie: modyfikacja / porzucanie / współtworzenie nowych planów

Korzyści zwinnych metodologii

- Biznes:
  - Brak długiej początkowej fazy analizy
  - Daje wymagania na bieżąco, na bazie tego co już widzi / użytkuje / opinii użytkowników
- Programiści:
  - Nie są rozliczani z estymat sprzed wielu miesięcy
  - Nie projektują szczegółowo modułów / klas / serwisów, które będę implementowane w odległej przyszłości i do tego czasu wymagania / interfejsy zupełnie się rozjadą

Scrum

- Zespół pracuje sprintami
  - Zwykle 1 – 2 tygodnie
  - Ustalenie zakresu, planowanie, realizacja
- Codzienne spotkania: dailies
  - Każdy członek zespołu opowiada co robił dzień wcześniej, jakie miał problemy, co robi dzisiaj
- Podsumowanie, prezentacja: Sprint Review
  - Prezentacja gotowego produktu wykonanego w bieżącym Sprincie

Scrum...

- Główne role:
  - Scrum Master
  - Product Owner
  - Developerzy
- Założenie o pełnej / dużej wymienności członków zespołu
  - W zespołach o dużym "rozwarstwieniu" w poziomach doświadczenia jest to fikcja
- Iteracyjne dopracowywanie procedur
- Backlog

Kanban

- Nacisk na przejrzyste zaprezentowanie obecnego stanu prac
- Tablica z zadaniami i ich etapami
  - Np: To do, In progress, Code review, In Test, Done
- W porównaniu do Scrum'a używany znacznie częściej w projektach głównie utrzymaniowych
- Backlog

A jak to wygląda w praktyce

- Czasem mówi się po prostu, że praca odbywa się "sprintami"
- Częścią wspólną wszelkich odmian jest w praktyce jedynie krótkie codzienne spotkanie
  - Chociaż odchodzi się niekiedy od tej "codzienności"
- Regularnie się zdarza, że Scrum Master'em jest jeden z programistów, Product Owner lub w praktyce jest tylko osoba prowadząca daily
- Scrum Scrum'owi nierówny :-)

A jak to wygląda w praktyce...

- Backlog
  - W idealnej sytuacji powinniśmy mieć tutaj zawsze co najmniej kilka dobrze opisanych historyjek, które są gotowe do "wzięcia"
- Wyceny zadań:
  - Małe / średnie / duże
  - 1, 2, 3, 5, 8, 13... (dosyć szybki punkt odcięcia)
  - Czasem jedynie priorytet bez wyceny
- Często dokładniejszy opis i wycena są robione na planowaniu sprintu

A jak to wygląda w praktyce...

- Kwestia spotkań
  - Obowiązkowo dailies, chocież też się zdarza pomijanie w czasie "stabilnych" faz
  - Planowanie
  - Prezentacja wyników sprintu: różna ranga
  - Retrospekcje: często pomijane lub o małej randze
  - Backlog grooming
- Pytanie co nazywamy Scrum'em?
  - Czy praca po prostu "sprintami" jest ok?

Pracownia

- Założyć / odświeżyć konto na Github
  - Kto nie pracował z Gitem koniecznie niech poczyta i wykona praktyczne ćwiczenia: założenie repozytorium, commit / submit, ściągnięcie repozytorium
- Kompletowanie 4-osobowych zespołów
- Jeśli chodzi o środowisko developerskie to polecam IntelliJ IDEA, bezpłatna Community Edition jest w zupełności wystarczająca na Nasze potrzeby