Programowanie zespołowe

Laboratorium 4 - modele tworzenia oprogramowania, manifest Agile i wstęp do Scruma

mgr inż. Krzysztof Szwarc

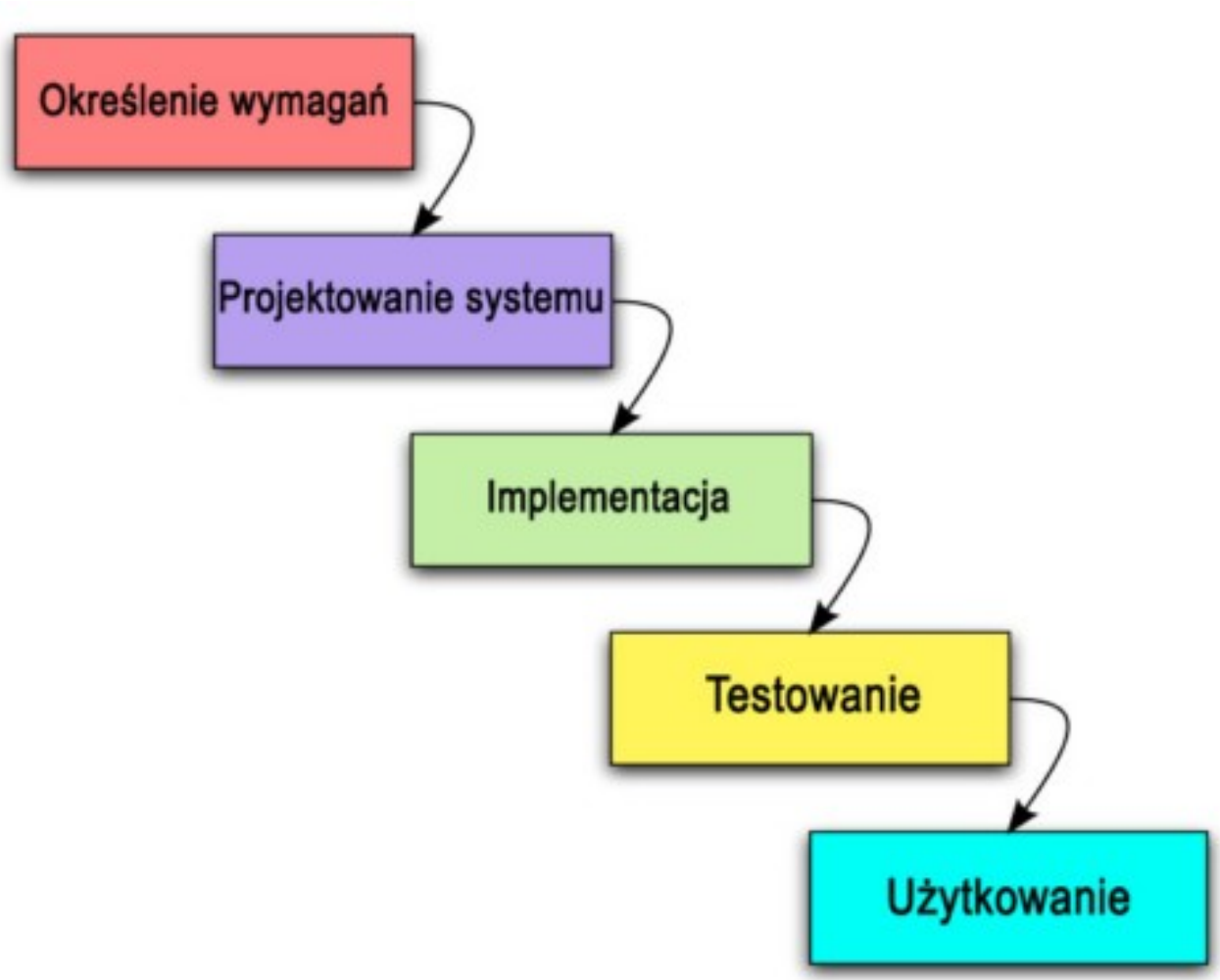
krzysztof@szwarc.net.pl

Sosnowiec, 14 marca 2017

Fazy procesu produkcji oprogramowania

1. Specyfikacja wymagań (określenie i ustalenie wymagań, które musi spełniać oprogramowanie).
2. Projektowanie (ustalenie architektury systemu i wymagań dla jego poszczególnych części).
3. Implementacja.
4. Integracja (połączenie składowych elementów w jeden system i jego testowanie).
5. Ewolucja (uruchomienie systemu, usuwanie znalezionych podczas użytkowania błędów i rozszerzanie systemu).

Model kaskadowy (waterfall)



http://zasoby.open.agh.edu.pl/ 10sdczerner/page/model_kaskadowy.html

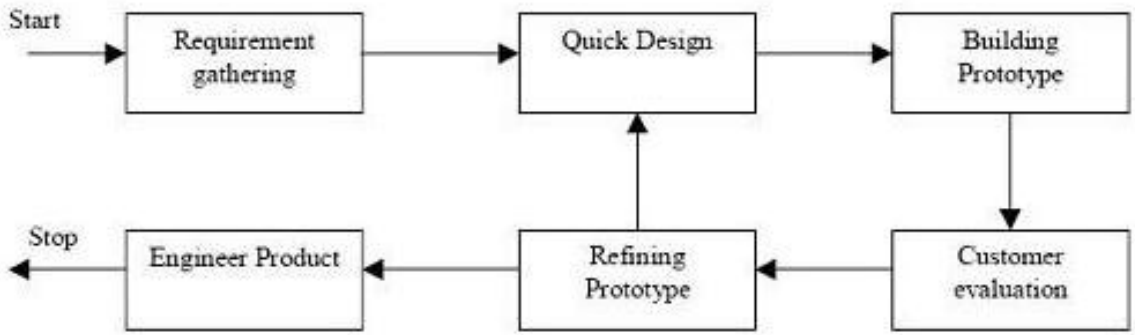
Cechy modelu kaskadowego

- Każda następna faza rozpoczyna się dopiero po zakończeniu fazy poprzedzającej.
- Jeśli faza zwróci niesatysfakcjonujący produkt cofamy się wykonując kolejne iteracje do czasu otrzymania satysfakcjonującego produktu.
- Łatwe planowanie i zarządzanie wykonaniem (poprzez identyfikację i uporządkowanie procesu tworzenia oprogramowania).
- Wysoki koszt iteracji spowodowany powtarzaniem wielu czynności.
- Stosowany, gdy wymagania są zrozumiałe i przejrzyste.

Model prototypowy

Fazy

- Tworzenie podczas projektowania prototypu.
- Po jego akceptacji przechodzi się do kolejnych etapów.



Prototyping Model

http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/

Cechy modelu prototypowego

- Prototypowanie zapobiega błędnemu zrozumieniu wymagań.
- Prototyp jest łatwy do zmiany.
- Umożliwia rozpoczęcie szkolenie obsługi oprogramowania po stronie klienta.
- Klient uznaje, że aplikacja jest prawie gotowa, a w rzeczywistości jest ona dopiero w początkowej fazie rozwoju.

Popularne rodzaje prototypów

- Rozpisanie lub narysowanie GUI na kartce papieru.
- Implementacja części modułów.
- Użycie narzędzi typu RAD (Rapid application development).
- Implementacja metod działających tylko dla części przypadków.

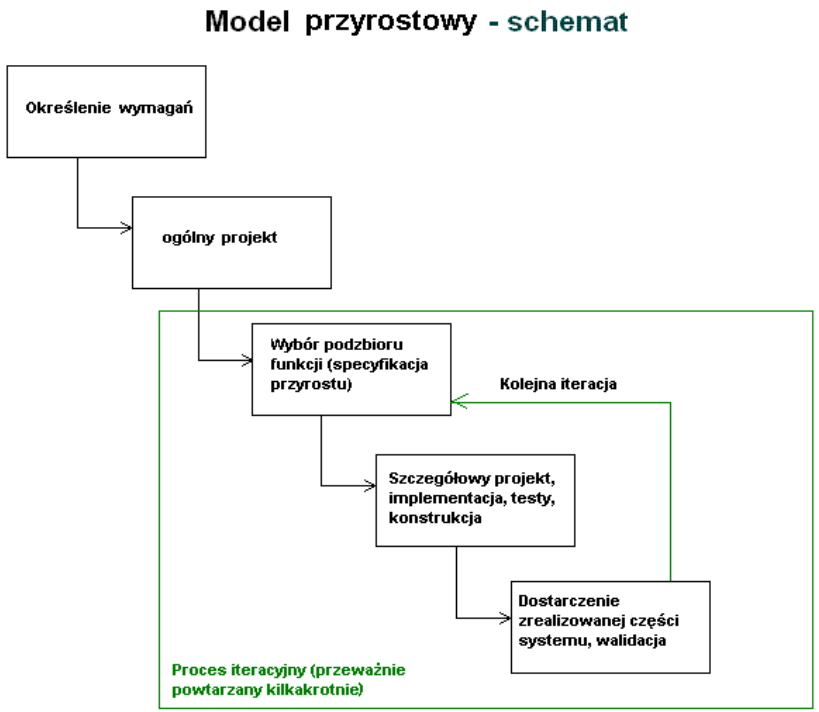
Zadanie

1. Korzystając z dowolnego narzędzia RADowego przygotuj panel logowania dla użytkownika.

Model przyrostowy

Fazy

- Określenie wymagań.
- Wybór podzbioru funkcji systemu.
- Projekt i realizacja podzbioru.
- Testowanie fragmentu i dostarczenie go.
- Powtórzenie kroków do zakończenia prac nad systemem.



http://lucas-informatyka.wdfiles.com/local–files/psiegzamin/rys7.GIF/

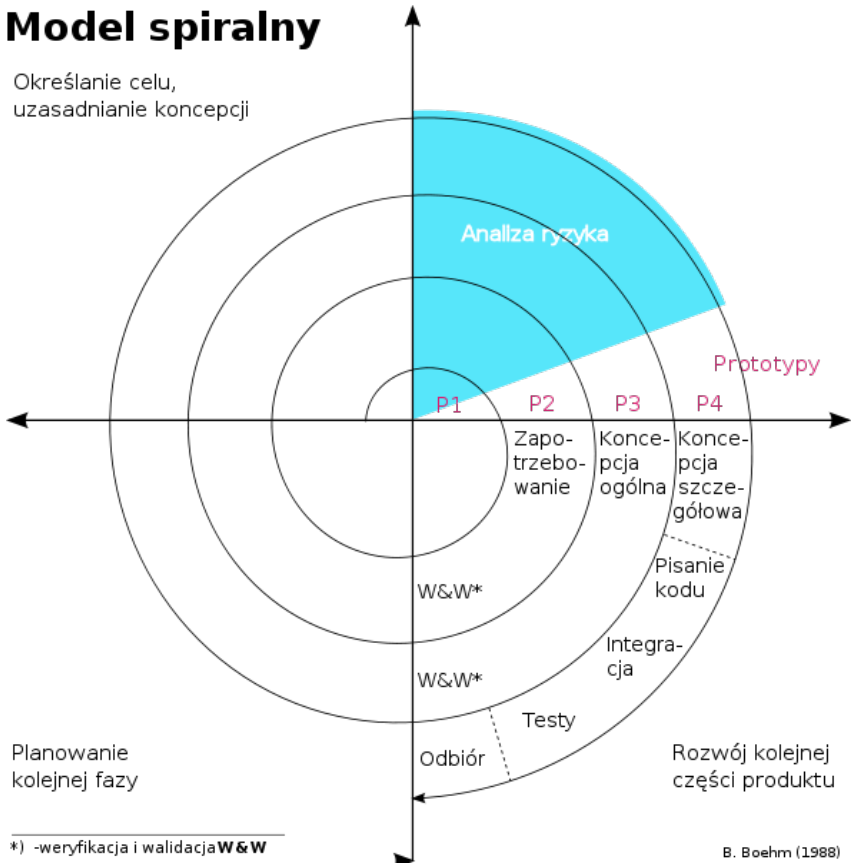
Cechy modelu przyrostowego

- Częste kontakty z klientem.
- Brak konieczności zdefiniowania wszystkich wymagań od razu.
- Umożliwia rozpoczęcie szkolenie obsługi oprogramowania po stronie klienta.
- Elastyczne reagowanie na opóźnienia (przyspieszenie prac nad innym fragmentem).
- Dodatkowe koszty.
- Trudności z wydzieleniem podzbioru niezależnych funkcji.

Model spiralny

Spirala

- Każda pętla reprezentuje jedną fazę procesu.
- Najbardziej wewnętrzna pętla przedstawia początkowe etapy projektowania.
- Wyróżnia się cztery etapy, przez które przechodzą poszczególne fazy (planowanie, analiza ryzyka, realizacja i zatwierdzenie oraz ocena).

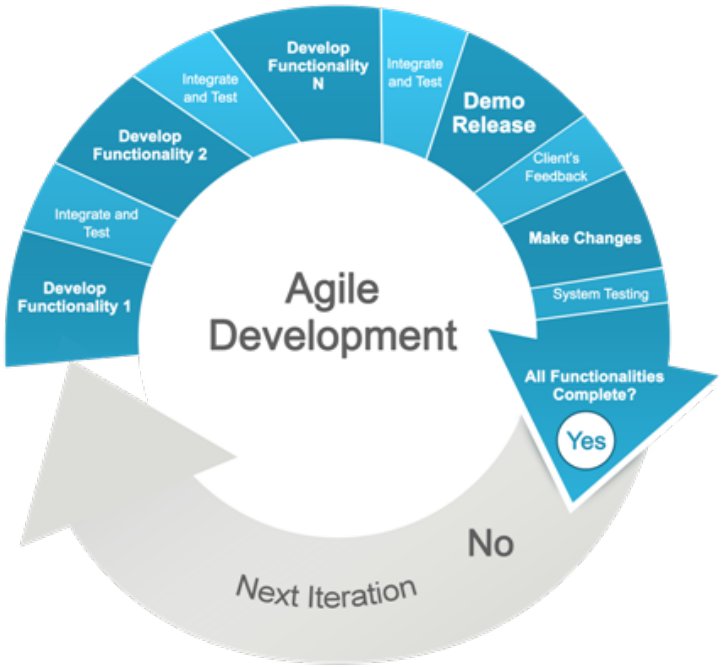


https://pl.wikipedia.org/wiki/

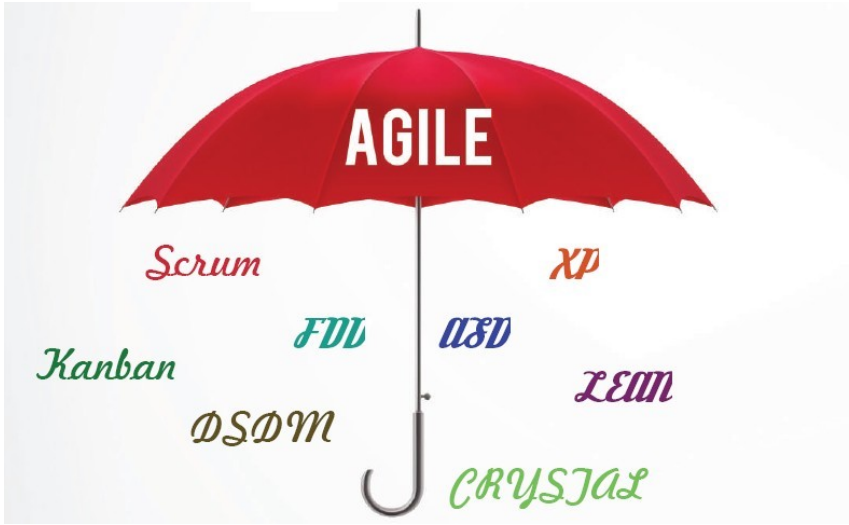
Cechy modelu spiralnego

- Elastyczny (jeden cykl spirali może przebiegać w oparciu o model kaskadowy, a drugi skorzystać z prototypowania).
- Faza oceny pozwala uniknąć błędów oraz wykryć je we wczesnym etapie.
- Istnieje możliwość rozwijania projektu.
- Wymaga doświadczenia i wiedzy.
- Wysoki koszt usuwania błędów wykrytych w końcowych etapach projektu.

Programowanie zwinne (agile programming)



http://varunm.com



http://pro-tekconsulting.com

Cechy programowania zwinnego

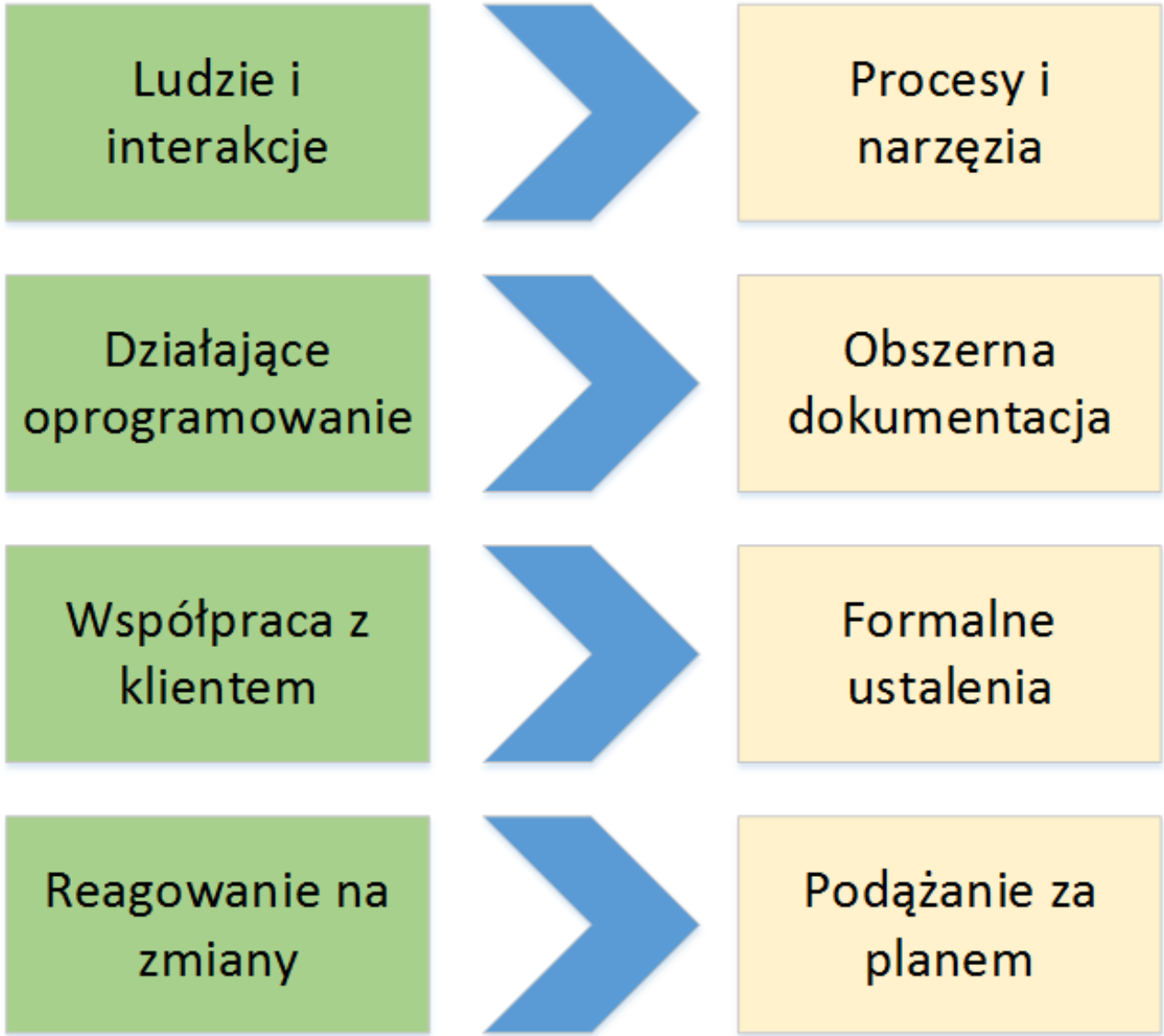
- Szybka reakcja na zmiany.
- Ciągły kontakt z klientem.
- Wartościowy produkt po każdym cyklu.
- Testy wykonywane na bieżąco.
- Stworzony produkt końcowy może różnić się od początkowych założeń.
- Wymagany dobry zespół.

Agile

Manifest Agile jest deklaracją wspólnych zasad dla zwinnych metodyk tworzenia oprogramowania (m.in. scrum, programowanie ekstremalne i Feature Driven Development).

https://svsg.co/

Założenia manifestu



Zasady uzupełniające

1. Najważniejsze jest zadowolenie klienta osiągane dzięki wczesnemu i ciągłemu wdrażaniu wartościowego oprogramowania.
2. Trzeba być elastycznym na zmiany wymagań nawet na późnym etapie jego rozwoju. Procesy zwinne wykorzystują zmiany dla zapewnienia klientowi konkurencyjności.
3. Należy dostarczać funkcjonujące oprogramowanie często (w kilkutygodniowych lub kilkumiesięcznych odstępach).
4. Zespoły biznesowe i deweloperskie muszą ściśle ze sobą współpracować (przez cały czas trwania projektu).
5. Należy motywować ludzi - zapewnić im potrzebne środowisko, wsparcie i zaufanie.
6. Najefektywniejsza jest rozmowa twarzą w twarz (przepływ informacji).

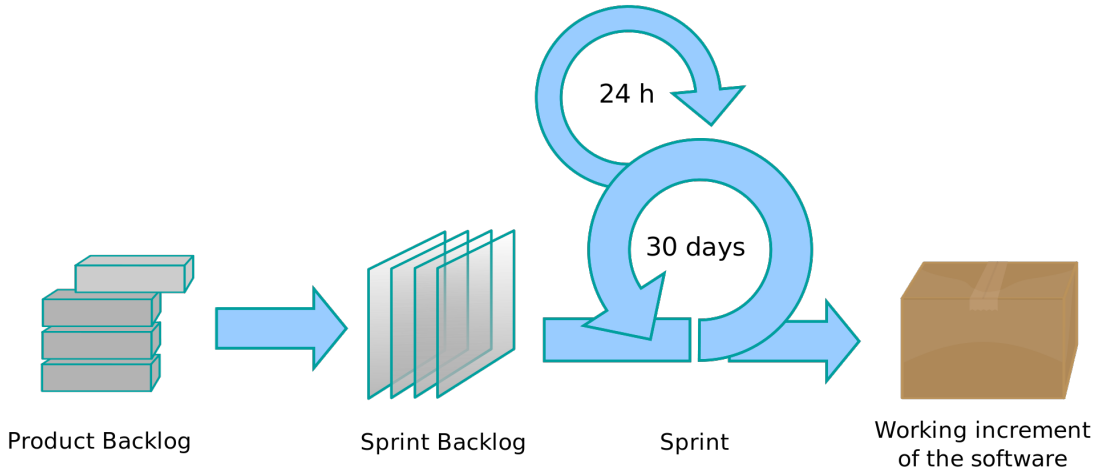
Zasady uzupełniające cz. 2

7. Działające oprogramowanie jest podstawową miarą postępu.
8. Procesy zwinne umożliwiają zrównoważony rozwój (sponsorzy, deweloperzy i użytkownicy utrzmują równe tempo pracy).
9. Skupienie na technicznej doskonałości i dobrym projektowaniu zwiększa zwinność.
10. Prostota jest najważniejsza.
11. Ważna jest samoorganizacja zespołów (samodzielnie podejmowanie decyzji, zarządzanie i organizowanie pracy).
12. Ciągła poprawa (w regularnych odstępach następuje analiza wydajności i podejmowanie działań ją zwiększających).

Scrum

Scrum

Scrum jest jedną z metodyk zwinnych zgodnych z manifestem Agile. Zakłada on pracę zespołu przez określony okres czasu (sprint), której efektem ma być dostarczenie kolejnej działającej wersji produktu.



http://www.marcinmuras.com/

Zadanie

1. Załóż konto na `https://play.planningpoker.com`

Dziękuję za uwagę