



AGH

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**

AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Reguły Agile & podstawy Scrum

Piotr Boryło

Plan wykładu

- » Agile - historia i manifest
- » Podstawy Agile
- » Podstawy Scrum
 - Role w zespole
 - Czynności
 - Artefakty

Historia Agile

- » 2001 r., Utah
- » 17 przedstawicieli dużych firm IT wymieniło poglądy na temat metod skutecznego tworzenia oprogramowania
- » Wcześniej niezależnie pracowali nad/w Scrum, DSDM czy XP

Historia Agile

- » Nazwa Agile powstała w 2001
- » Najbardziej znane konkretne metody w latach 1980-90
- » Agile to tak naprawdę wspólny mianownik dla czegoś to już istniało
- » Nazwa mogłaby zniknąć
- » Agile – *the way we do software*

Efekty spotkania

- » The Agile Manifesto – zwerbalizowanie wartości i celów
- » The Agile Principles – 12 zasad i wskazówek dla użytkowników Agile
- » The Agile Alliance – społeczność

The Agile Manifesto

» Dokument inicjujący zmianę

We are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:

The Agile Manifesto

- » Individuals and interactions over processes and tools
- » Working software over comprehensive documentation
- » Customer collaboration over contract negotiation
- » Responding to change over following a plan

The Agile Manifesto

That is, while there is value in the items on the right, we value the items on the left more.

Individuals and interactions over processes and tools

- » Członkowie zespołu muszą mieć wystarczającą ilość czasu i muszą *grać* zespołowo
- » Przejrzysta i szybka komunikacja
- » Praca zespołowa i samoorganizacja
- » Zaangażowanie personalne
- » Chęć ciągłej poprawy
- » Satysfakcja

Working software over comprehensive documentation

- » Działający produkt spełniający potrzebę
- » Skończyły nam się pieniądze, czy możemy dostać ukończone 75% projektu:
 - Waterfall
 - Agile
- » Dokumentacja techniczna w Agile - TAK
- » Plany, raporty, szablony, dokumenty - NIE

Working software over comprehensive documentation



(working software)

OVER



Customer collaboration over contract negotiation

» Kiedyś:

- Start projektu: negocjacja kontraktu
- Wykonanie: negocjacja zmian
- Koniec: jeśli produkt nie spełnia wymagań kolejne etapy negocjacji zmian

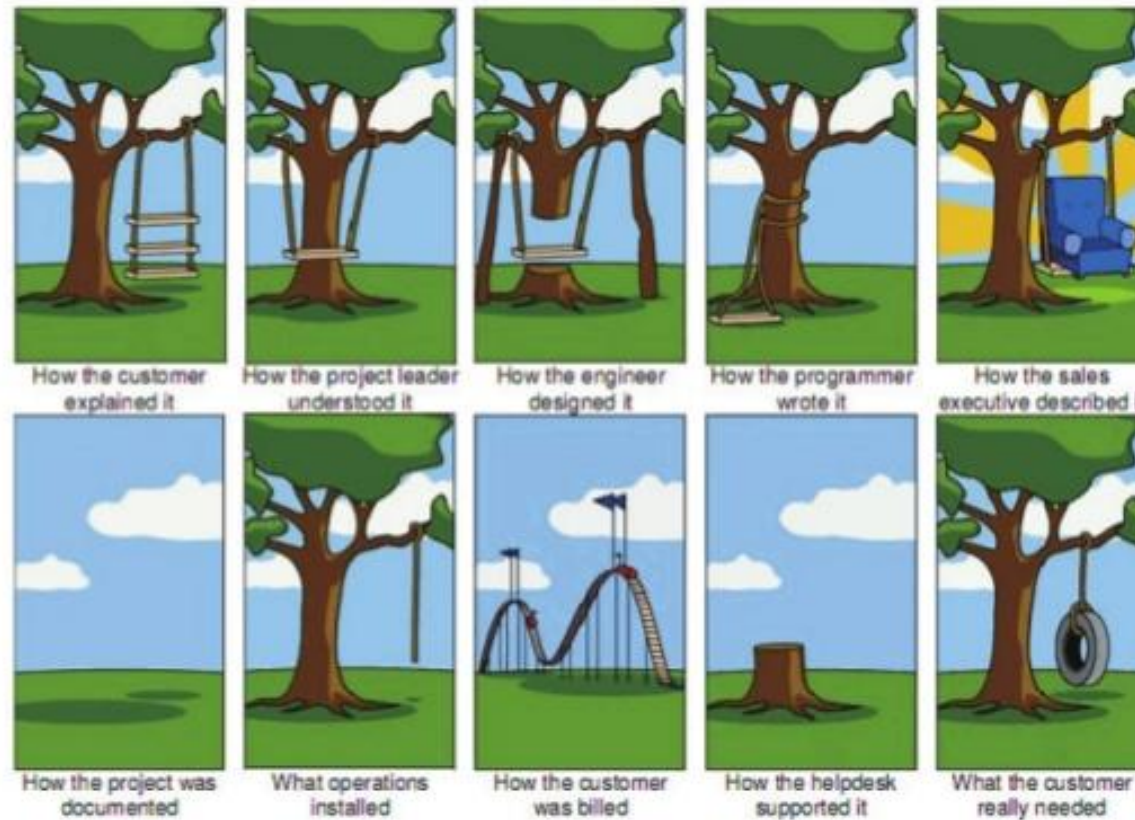
Customer collaboration over contract negotiation

- » Kluczem były negocjacje
- » Wartość stanowi współpraca i partnerstwo w celu:
 - Wzajemnego zrozumienia
 - Zrozumienia potrzeb
 - Rozwoju
- » Na początku projektu i klient i wykonawca mają najmniejszą wiedzę!

Responding to change over following a plan

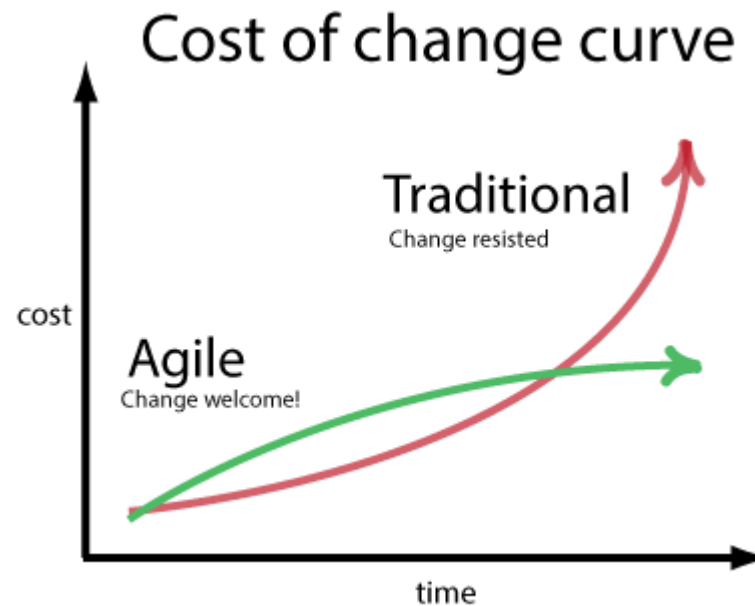
- » Szybkie sprzężenie zwrotne od klienta
- » Szybka reakcja na zmiany potrzeb klienta:
 - Klient w trakcie projektu zaczyna rozumieć czego chce
 - Zmienia się sytuacja na rynku
 - Zmienia się sytuacja w firmie
 - Zmienia się strategia biznesowa klienta

Responding to change over following a plan



<http://www.tamingdata.com/2010/07/08/the-project-management-tree-swing-cartoon-past-and-present/>

Responding to change over following a plan



The Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.

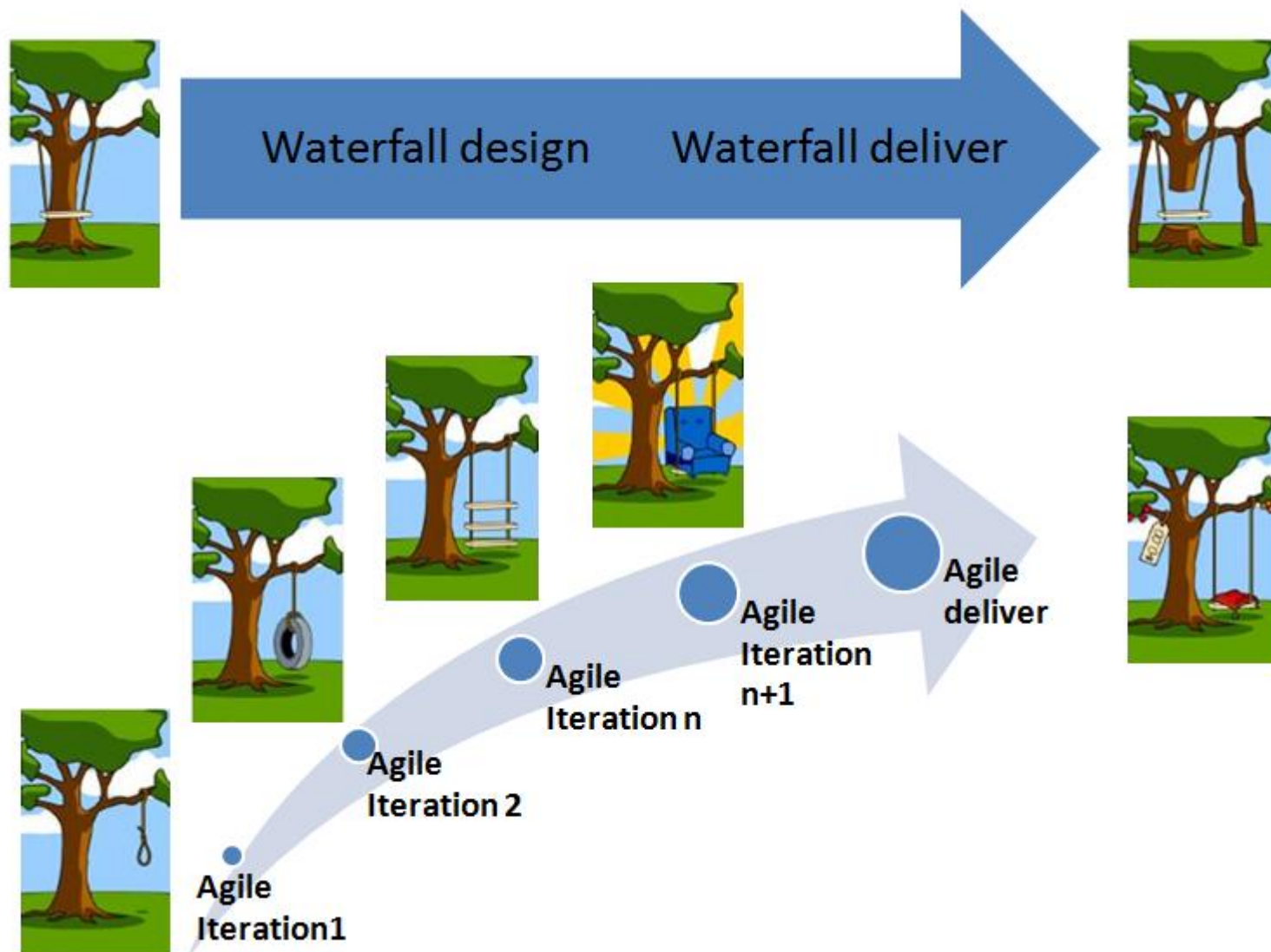
The Agile Principles

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The Agile Principles – podział

- » Satysfakcja klienta: 1,2,3,4
- » Jakość: 1,3,4,6,7,8,9
- » Praca zespołowa: 4,5,6,8,11,12
- » Zarządzanie projektem: 2,8,10

Zalety Agile - podsumowanie



Zalety Agile - podsumowanie

- » Wiedza na temat produktu rośnie w trakcie realizacji
- » Redukcja nieproduktywnych zadań: spotkania, e-mail, prezentacje, dokumenty
- » Jakość i szybkość w pracy zespołowej
- » Szybka odpowiedź na zmiany
- » Redukcja kosztów porażki i samofinansowanie

Podstawy Scrum

Struktura Scrum

- » Scrum nie jest procedurą postępowania a strukturą/ramą do organizacji i zarządzania pracą
- » Implementacja Agile

Struktura Scrum

- » Role w zespole
- » Czynności
- » Artefakty

Struktura Scrum

» Role w zespole

- Product Owner

Co zaimplementować i w jakiej kolejności

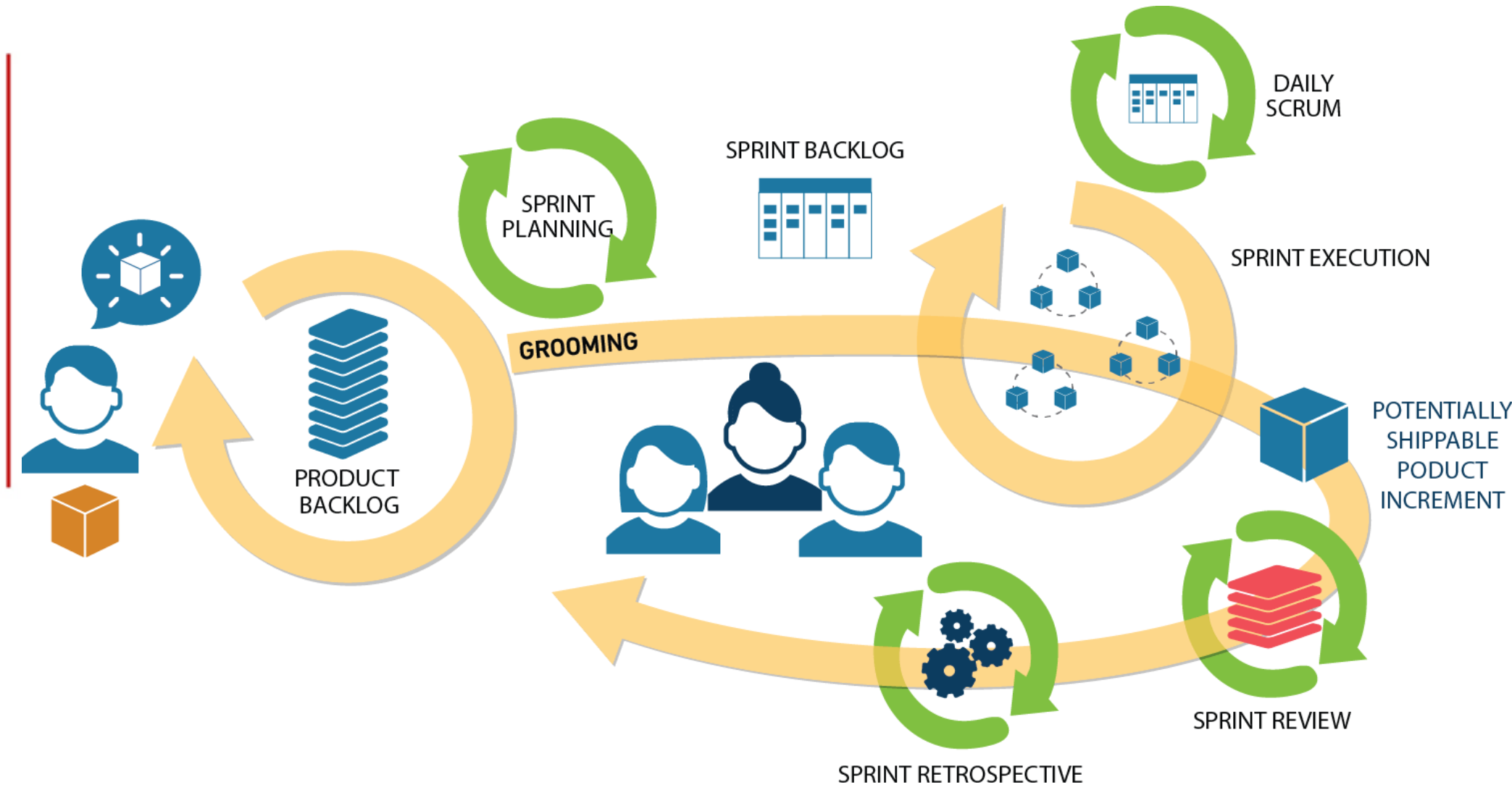
- Scrum Master

Przewodnictwo dla zespołu

- Zespół

Jak zaimplementować

Struktura Scrum



Struktura Scrum

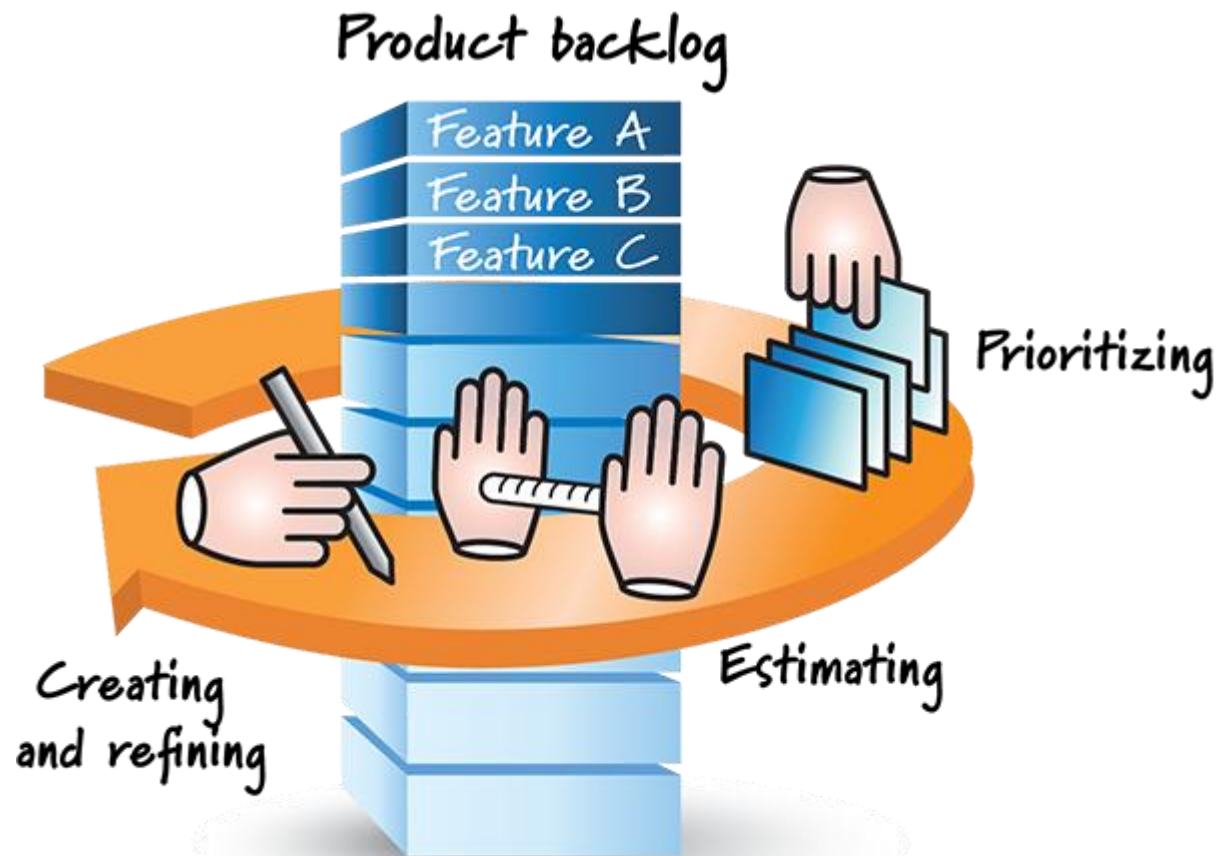
- » Artefakty:
 - *backlog* produktu
 - *backlog* iteracji (*sprintu*)
 - produkt o potencjale komercyjnym (przyrostowa rozbudowa)

Struktura Scrum

» Czynności:

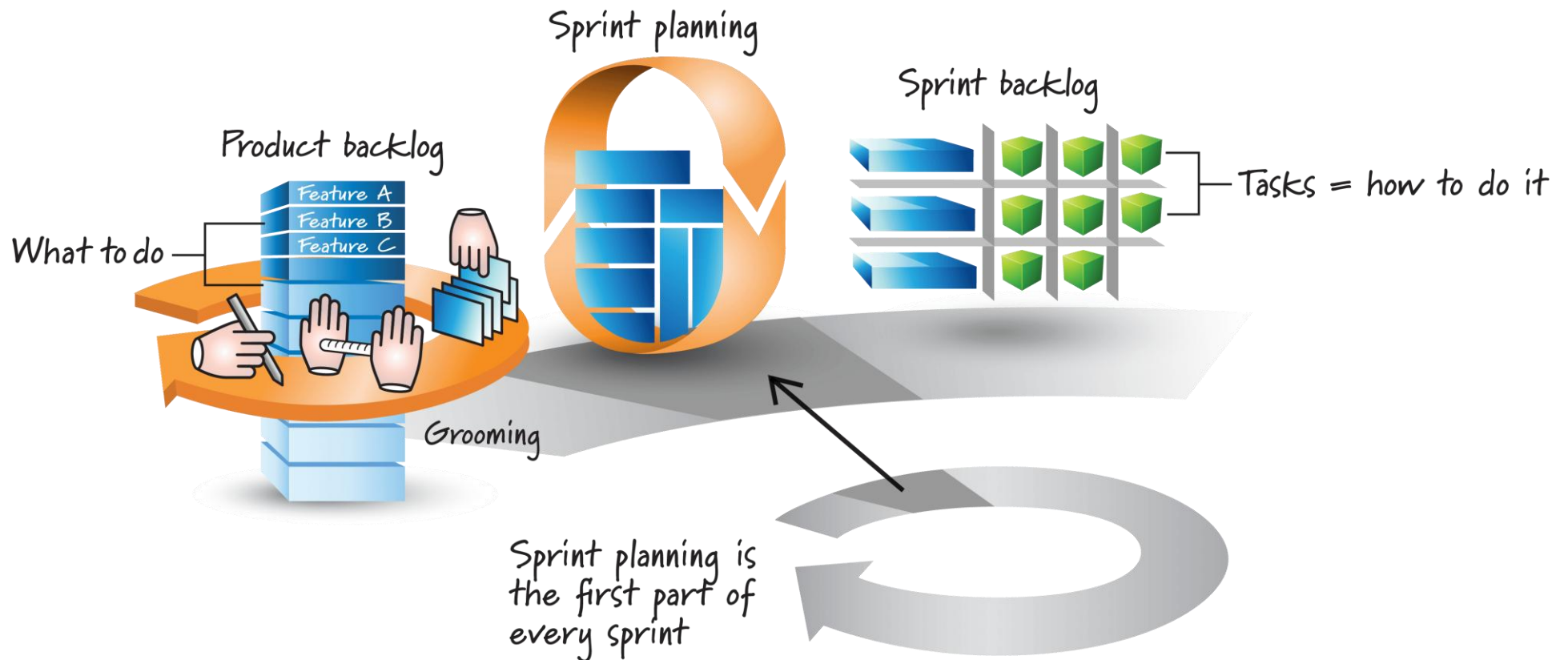
- Sprint
- Sprint planning
- Daily scrum
- Sprint Execution
- Sprint review
- Sprint retrospective
- Product backlog grooming

Grooming backlogu produktu

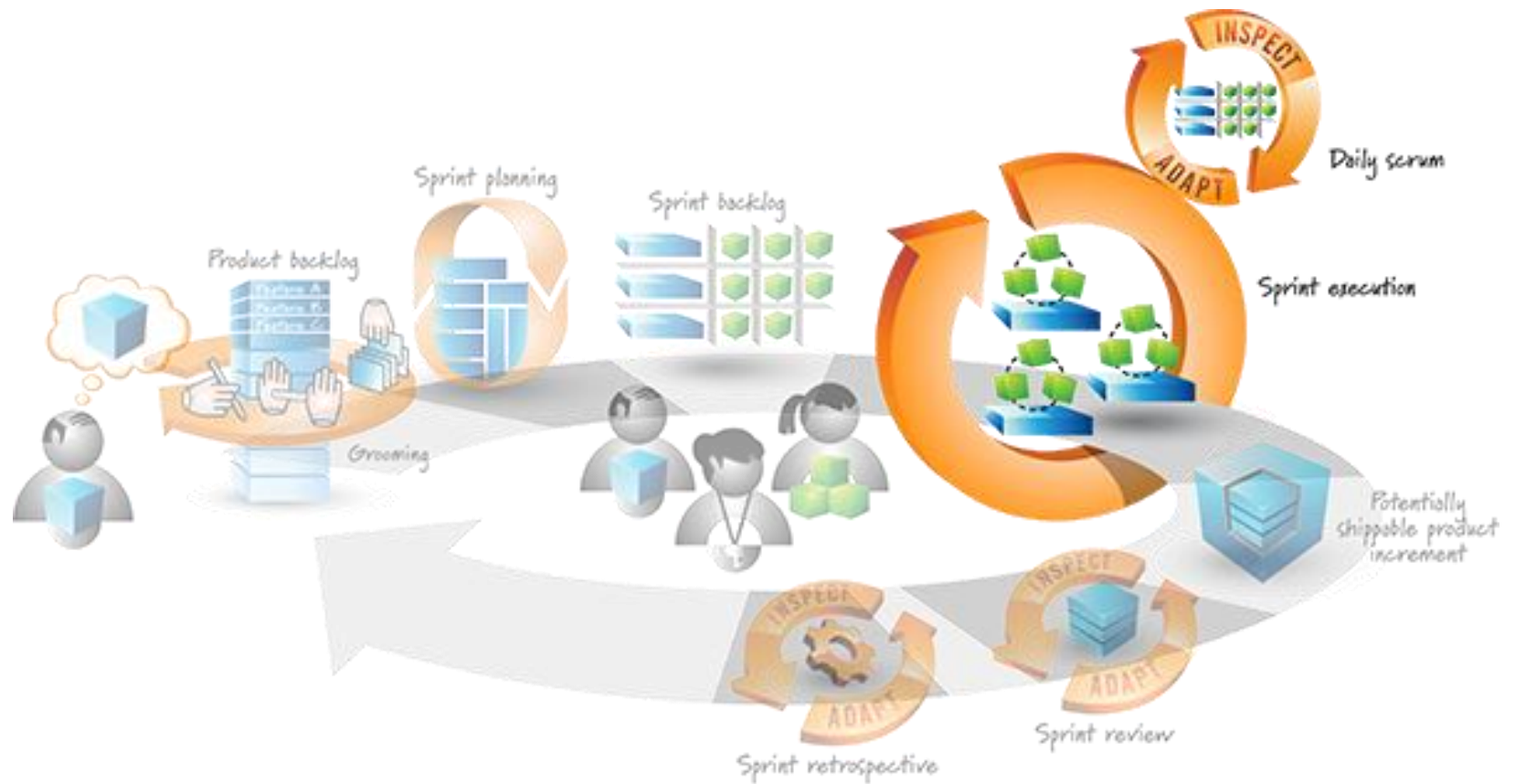


Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

Sprint planning



Daily Scrum



Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

Daily Scrum

- » Co zrobiłem w czasie ostatniego dnia
- » Czym będę się zajmował dzisiaj
- » Jakie są przeszkody i utrudnienia w osiąganiu postępów

Efekt pracy

» *Definition of done*

» Przykład minimalistyczny dla oprogramowania:

Zaprojektowany, stworzony,
zintegrowany, przetestowany i
udokumentowany element produktu
końcowego

» **Potentially** Shippable Product Increment

Zakończenie iteracji

- » Sprint review
 - Inspekcja i adaptacja produktu
 - Wszyscy zaangażowani
- » Sprint retrospective
 - Tylko dla zespołu, SM i PO
 - Inspekcja i adaptacja procesu
 - Co i jak należy poprawić

Sprint

- » Szkielet struktury Scrum
- » Iteracja/cykl
- » Maksymalnie 1 miesiąc, stała długość
- » Cel sprintu nie może zostać zmieniony w trakcie wykonania
- » Na koniec powinien spełnić definicję wykonania

Sprint

- » Zalety zamknięcia w ramach czasowych
 - Ustalony limit pracy w toku
 - Wymuszona priorytetyzacja
 - Widoczny progres
 - Unikanie zbędnego perfekcjonizmu
 - Motywacja do skończenia zadań
 - Lepsza przewidywalność

Sprint

- » Zalety krótkiego trwania
 - Łatwość planowania
 - Szybkie sprzężenie zwrotne
 - Ograniczenie rozbieżności z wymaganiami klienta
 - Poprawa zysku z inwestycji
 - Utrzymywanie ekscytacji

Sprint

- » Cel sprintu nie może zostać zmieniony po uzgodnieniu
- » Cel sprintu jest celem biznesowym i wyznacza jego wartość
- » Obustronna zgoda

Definition of done

TABLE 4.1 Example Definition-of-Done Checklist

Definition of Done	
<input type="checkbox"/>	Design reviewed
<input type="checkbox"/>	Code completed
<input type="checkbox"/>	Code refactored
<input type="checkbox"/>	Code in standard format
<input type="checkbox"/>	Code is commented
<input type="checkbox"/>	Code checked in
<input type="checkbox"/>	Code inspected
<input type="checkbox"/>	End-user documentation updated
<input type="checkbox"/>	Tested
<input type="checkbox"/>	Unit tested
<input type="checkbox"/>	Integration tested
<input type="checkbox"/>	Regression tested
<input type="checkbox"/>	Platform tested
<input type="checkbox"/>	Language tested
<input type="checkbox"/>	Zero known defects
<input type="checkbox"/>	Acceptance tested
<input type="checkbox"/>	Live on production servers

- » Dopasowane do zespołu i warunków
- » Co kiedy definicja nie jest spełniona na koniec sprintu?
- » *Done vs done-done*

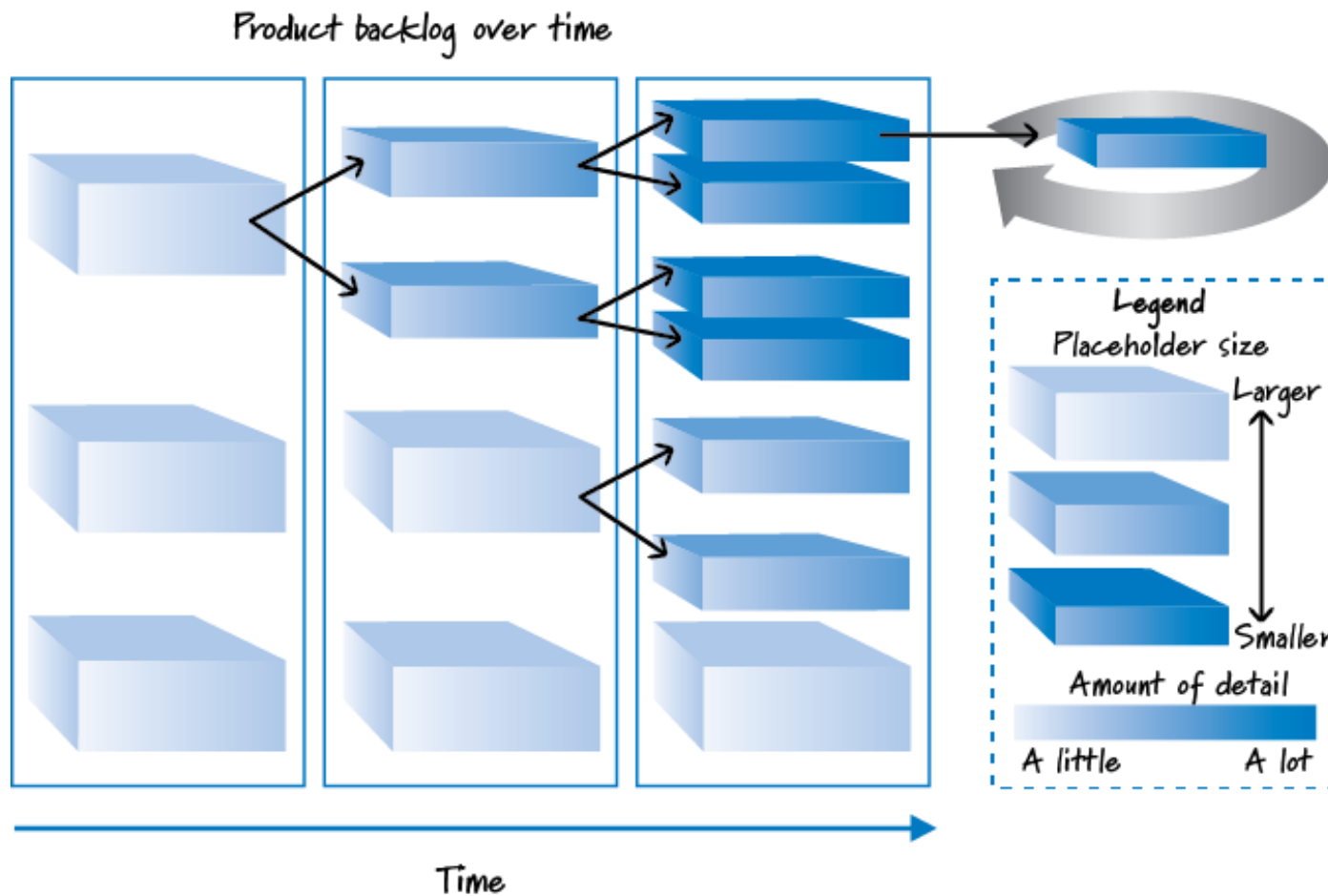
Wymagania użytkownika

- » Wymagania mogą się zmieniać
- » Historie użytkownika (*user stories*)

Przyjęty format opisu wartości biznesowych

- » Wymagania są wynikiem ciągłego dialogu a nie negocjacji i wielostronnicowej dokumentacji
- » *Product backlog items* – oddzielne wartości biznesowe

Wymagania użytkownika



Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

Wymagania użytkownika

» Historie użytkowników zrozumiałe dla biznesu i twórców produktu

– Karta:

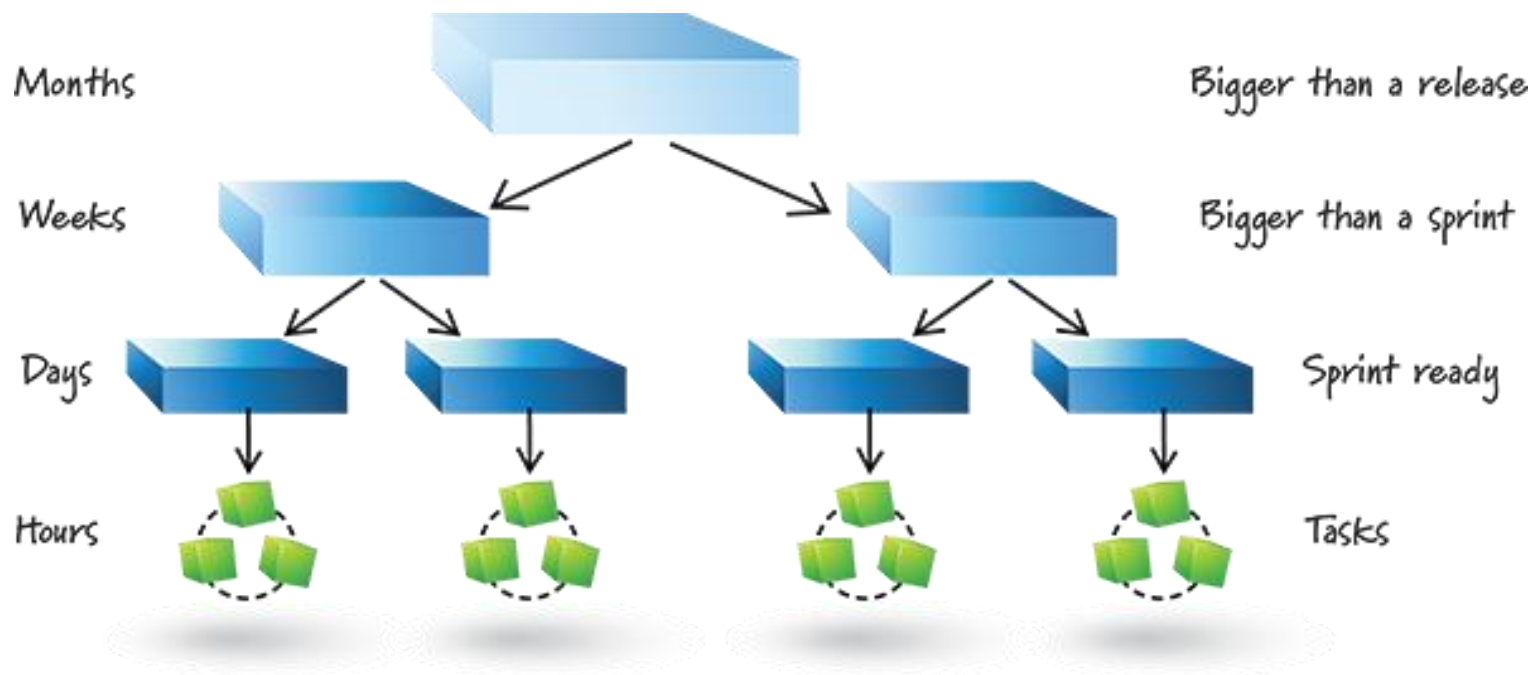
As a <user role> I want to <goal> so that <benefit>

– Dialog, zwykle długotrwały

– Potwierdzenie zrozumienia

– Różny poziom szczegółowości

Wymagania użytkownika



Wymagania użytkownika

» **INVEST** in Good Stories

- Independent
- Negotiable
- Valuable (also technical stories)
- Estimable
- Small
- Testable



Dziękuję za uwagę!

Piotr Boryło