Zwinne metodyki AGILE

Dr hab. inż. Ilona Bluemke

Oraz slajdy Ian Sommerville „Software Engineering” 10 ed

LOOP

***L*ate (późno)**

***O*ver budget (przekroczony budżet)**

***O*vertime (nadgodziny)**

***P*oor quality (kiepska jakość)**

Manifest zwinności (*Agile*)

Luty 2001

Kent Beck (karty CRC, xUnit, XP)

Alistair Cockburn (przypadki użycia)

Marin Fowler (refaktoryzacja, UML Distilled)

Jim Highsmith (Adaptive Software Development)

Ważniejsze jednostki i interakcje niż procesy i narzędzia

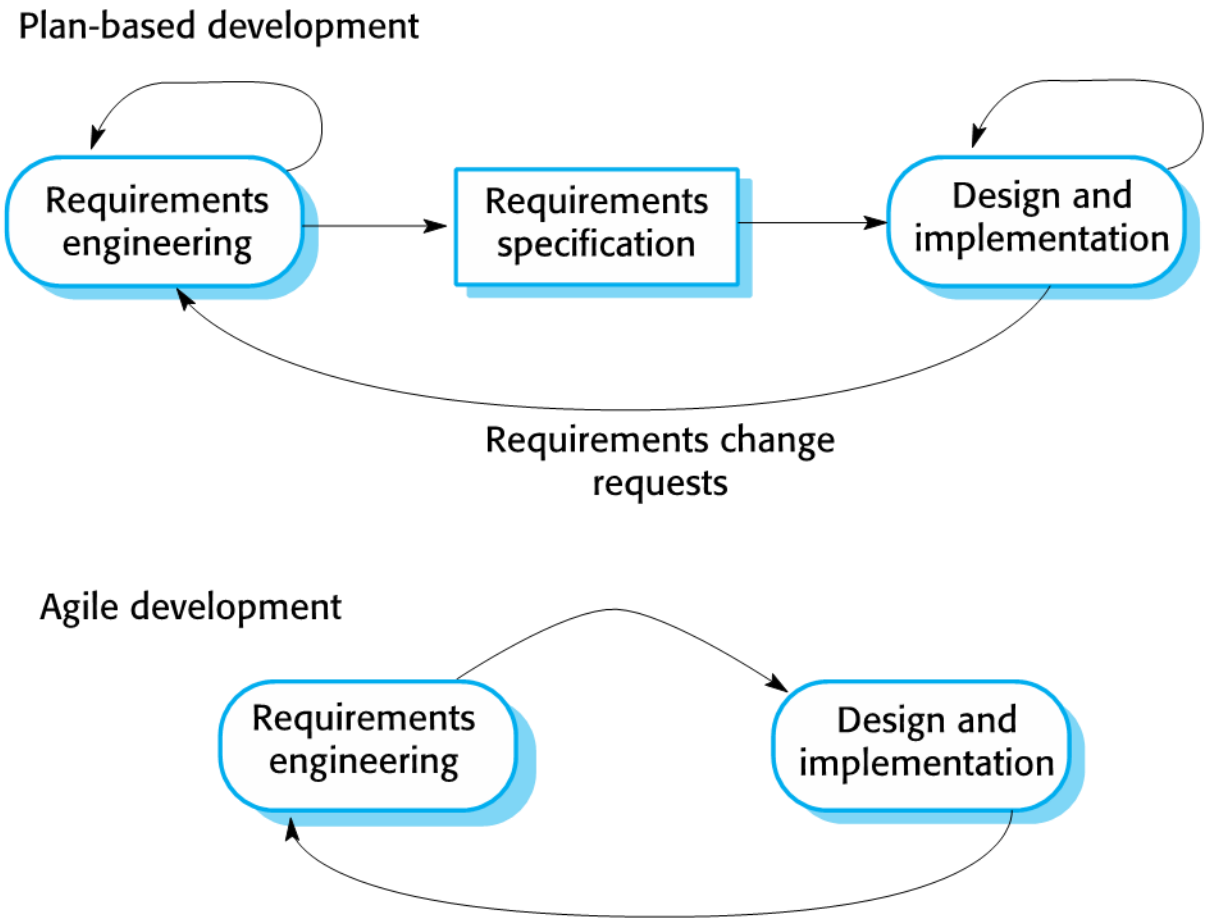
ISO 9001:2000

główne części:

- System zarządzania jakością (dokumentacja)
- Odpowiedzialność kierownictwa
- Zarządzanie zasobami
- Realizacja wyrobu
- Pomiary, analiza i doskonalenie

Problem – zbyt dużo dokumentacji, brak elastyczności

Źródło - Ian Sommerville



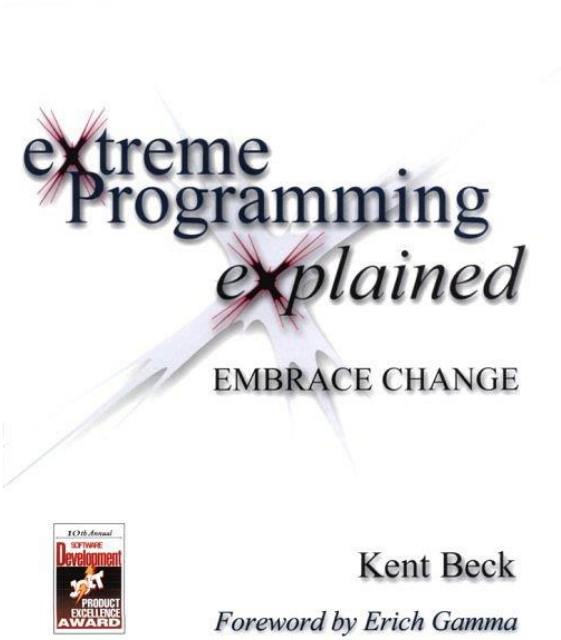
Manifest zwinności

Ważniejsze:

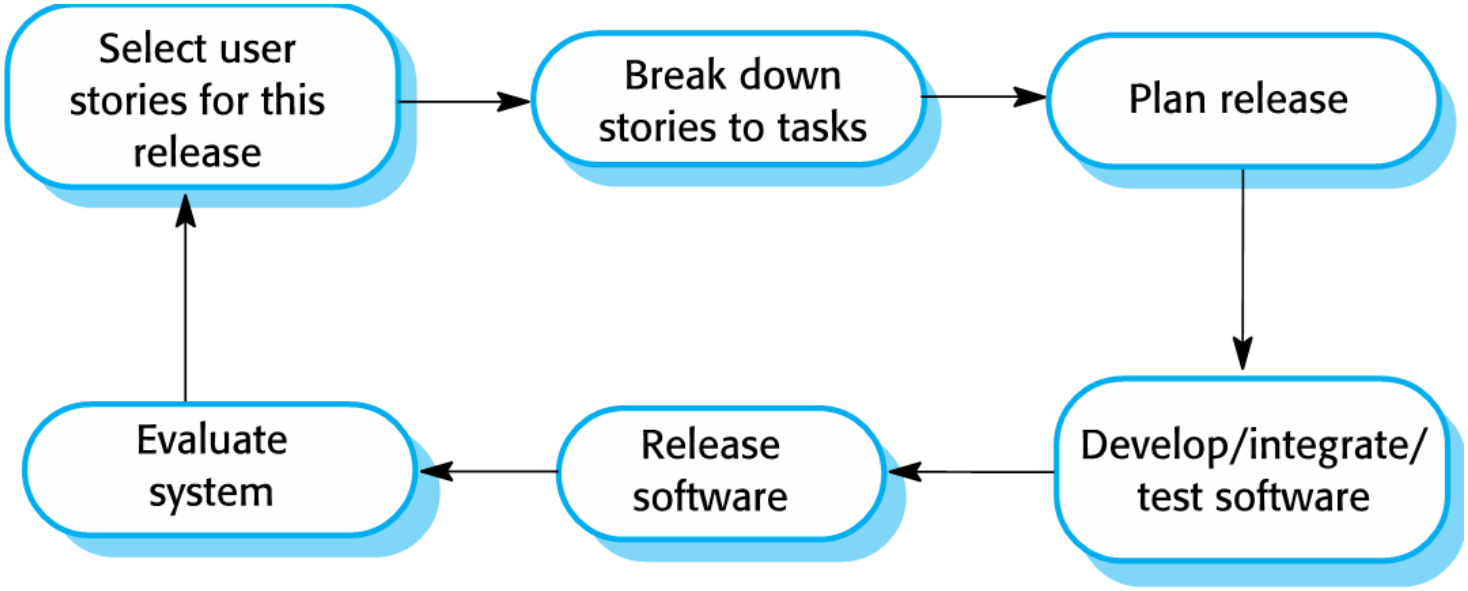
- **Jednostki i interakcje niż procesy i narzędzia**
- **Działające oprogramowanie niż obszerna dokumentacja**
- **Współpraca klienta niż negocjacja kontraktu**
- **Nadążanie za zmianami niż trzymanie się planu**

Programowanie Ekstremalne (XP)

- lekka (zwinna)
- metodyka tworzenia oprogramowania



Źródło - Ian Sommerville



Wybrane praktyki XP

- Klient na miejscu
- Krótkie przyrosty i wydania
- Najpierw przypadki testowe potem kod
- Automatyzacja wykonywania testów
- Dokumentacja = Przypadki testowe + kod
- Programowanie parami
- Małe zespoły

Źródło - Ian Sommerville

Extreme programming practices (a)

Principle or practice	Description
Incremental planning	Requirements are recorded on story cards and the stories to be included in a release are determined by the time available and their relative priority. The developers break these stories into development 'Tasks'. See Figures 3.5 and 3.6.
Small releases	The minimal useful set of functionality that provides business value is developed first. Releases of the system are frequent and incrementally add functionality to the first release.
Simple design	Enough design is carried out to meet the current requirements and no more.
Test-first development	An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented.
Refactoring	All developers are expected to refactor the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.

Źródło - Ian Sommerville

Extreme programming practices (b)

Pair programming	Developers work in pairs, checking each other's work and providing the support to always do a good job.
Collective ownership	The pairs of developers work on all areas of the system, so that no islands of expertise develop and all the developers take responsibility for all of the code. Anyone can change anything.
Continuous integration	As soon as the work on a task is complete, it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.
Sustainable pace	Large amounts of overtime are not considered acceptable as the net effect is often to reduce code quality and medium term productivity
On-site customer	A representative of the end-user of the system (the customer) should be available full time for the use of the XP team. In an extreme programming process, the customer is a member of the development team and is responsible for bringing system requirements to the team for implementation.

Programowanie parami

- 2 osoby przy jednym komputerze
- Pary się zmieniają
- Wymiana wiedzy, informacji, mniejsze ryzyko
- efektywne

Raport Sackmana, Eriksona i Granta

Różnice w wydajności programowania jak **10:1**

Różnice w rozmiarze programu jak

5:1

Scrum

- Scrum is an agile method that focuses on managing iterative development rather than specific agile practices.
- There are three phases in Scrum.
  - The **initial** phase is an outline planning phase where you establish the general objectives for the project and design the software architecture.
  - This is followed by a series of **sprint cycles**, where each cycle develops an increment of the system.
  - The project **closure** phase wraps up the project, completes required documentation such as system help frames and user manuals and assesses the lessons learned from the project.

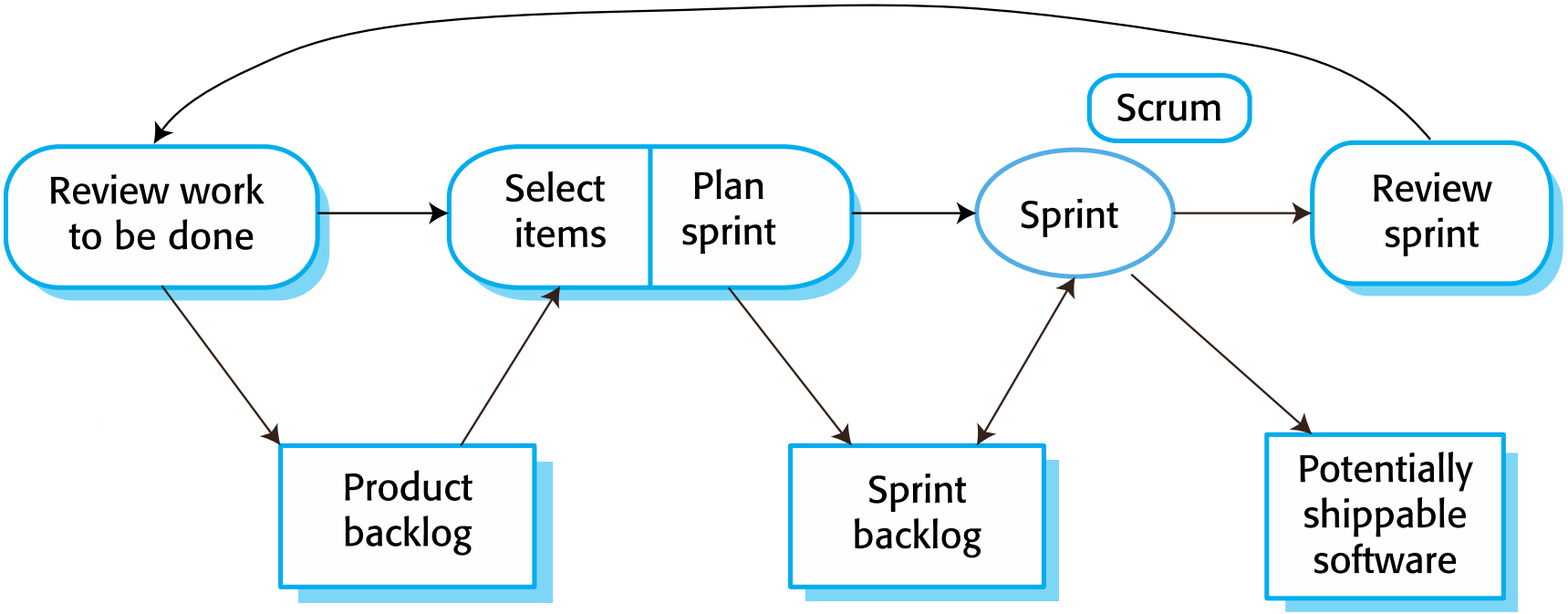
Scrum terminology (a)

Scrum term	Definition
Development team	A self-organizing group of software developers, which should be no more than 7 people. They are responsible for developing the software and other essential project documents.
Potentially shippable product increment	The software increment that is delivered from a sprint. The idea is that this should be 'potentially shippable' which means that it is in a finished state and no further work, such as testing, is needed to incorporate it into the final product. In practice, this is not always achievable.
Product backlog	This is a list of 'to do' items which the Scrum team must tackle. They may be feature definitions for the software, software requirements, user stories or descriptions of supplementary tasks that are needed, such as architecture definition or user documentation.
Product owner	An individual (or possibly a small group) whose job is to identify product features or requirements, prioritize these for development and continuously review the product backlog to ensure that the project continues to meet critical business needs. The Product Owner can be a customer but might also be a product manager in a software company or other stakeholder representative.

Scrum terminology (b)

Scrum term	Definition
Scrum	A daily meeting of the Scrum team that reviews progress and prioritizes work to be done that day. Ideally, this should be a short face-to-face meeting that includes the whole team.
ScrumMaster	The ScrumMaster is responsible for ensuring that the Scrum process is followed and guides the team in the effective use of Scrum. He or she is responsible for interfacing with the rest of the company and for ensuring that the Scrum team is not diverted by outside interference. The Scrum developers are adamant that the ScrumMaster should not be thought of as a project manager. Others, however, may not always find it easy to see the difference.
Sprint	A development iteration. Sprints are usually 2-4 weeks long.
Velocity	An estimate of how much product backlog effort that a team can cover in a single sprint. Understanding a team's velocity helps them estimate what can be covered in a sprint and provides a basis for measuring improving performance.

Scrum sprint cycle



The Scrum sprint cycle

- Sprints are fixed length, normally 2–4 weeks.
- The starting point for planning is the product backlog, which is the list of work to be done on the project.
- The selection phase involves all of the project team who work with the customer to select the features and functionality from the product backlog to be developed during the sprint.

Słabe strony XP

- **Brak dokumentacji**
- **Jeden klient (na miejscu)**
- **Zbyt krótka perspektywa planu**

Problemy metod zwinnych

- Trudność skalowania