

Wdrażanie i pielęgnacja oprogramowania

Wykładowca

Dr inż. Zofia Kruczkiewicz

Wdrażanie oprogramowania

Czym jest wdrażanie oprogramowania ?

<http://www.inf.ed.ac.uk/teaching/courses/inf2c-se/Lectures/deployment.pdf>

- Wdrażanie oprogramowania to wszystkie czynności, które po wykonaniu oprogramowania, umożliwiają przekazanie go do użytkownika np:
 - Tworzenie pakietów instalacyjnych
 - Wykonanie dokumentacji – np instrukcji instalacji itp.
 - Instalacja oprogramowania itp
 - Tworzenie kopii zapasowych, dostarczenie licencji...
- Ponad 50% zamówionego oprogramowania nie jest używana, ponieważ nie powiedzie się etap jego wdrożenia.
- 80% kosztów wytwarzania oprogramowania obejmuje proces jego wdrożenia i utrzymania

Metody wdrażania są zależne od dziedziny zastosowania oprogramowania (wykład 1)

1. Oprogramowanie systemowe
2. Systemy czasu rzeczywistego
3. Systemy informacyjne dla przedsiębiorstw
4. Oprogramowanie inżynierskie i naukowe
5. Systemy wbudowane
6. Oprogramowanie komputerów osobistych
7. Oprogramowanie internetowe
8. Sztuczna inteligencja
9. **Systemy hybrydowe (2, 4, 5, 7, 8)**
np Zintegrowany laboratoryjny system informatyczny do zarządzania informacjami medycznymi – LIMS (Laboratory Information Management System) oraz Laboratoryjny system informatyczny -LIS (Laboratory Information System)

Kluczowe problemy związane z wdrażaniem

<http://www.inf.ed.ac.uk/teaching/courses/inf2c-se/Lectures/deployment.pdf>

- **Procesy biznesowe** - większość dużych systemów programowania wymaga zmiany sposobu pracy przez użytkowników.
- **Trening** - nie ma sensu wdrażania oprogramowania, jeśli klienci nie mogą z niego korzystać.
- **Wsparcie** - jest konieczne
- **Wdrożenie** - jak fizycznie wdrożyć oprogramowanie
- **Sprzęt** - czy sprzęt klienta jest gotowy do pracy?
- **Specjaliści IT** - czy użytkownik ma wiedzę informatyczną do zainstalowania i obsługi oprogramowania?
- **Aktualizacja** - nie można jej uniknąć!
- **Integracja** - czy wykonany program współpracuje poprawnie z innymi systemami klienta.
- **Wydajność** – czy użytkownik posiada inny sprzęt niż używany w procesie wytwarzania oprogramowania

Wersja 1.0 wdrożenia

Instalacja

- Oprogramowanie powinno być łatwe do zainstalowania
- Różne aplikacje dla różnych platform
 - **desktopowe** (pliki wykonywalne, pakiety instalacyjne itd.)
 - **internetowe** (gdzie / jak będzie hostowany ?, itp.)
 - **mobilne** (wewnętrzna aplikacja biznesowa, sklep z aplikacjami, Google Play, ...)
 - **lub ich połączenie**

Wsparcie

– Jak rejestrować i obsługiwać zgłoszenia serwisowe od klientów

Pielęgnacja

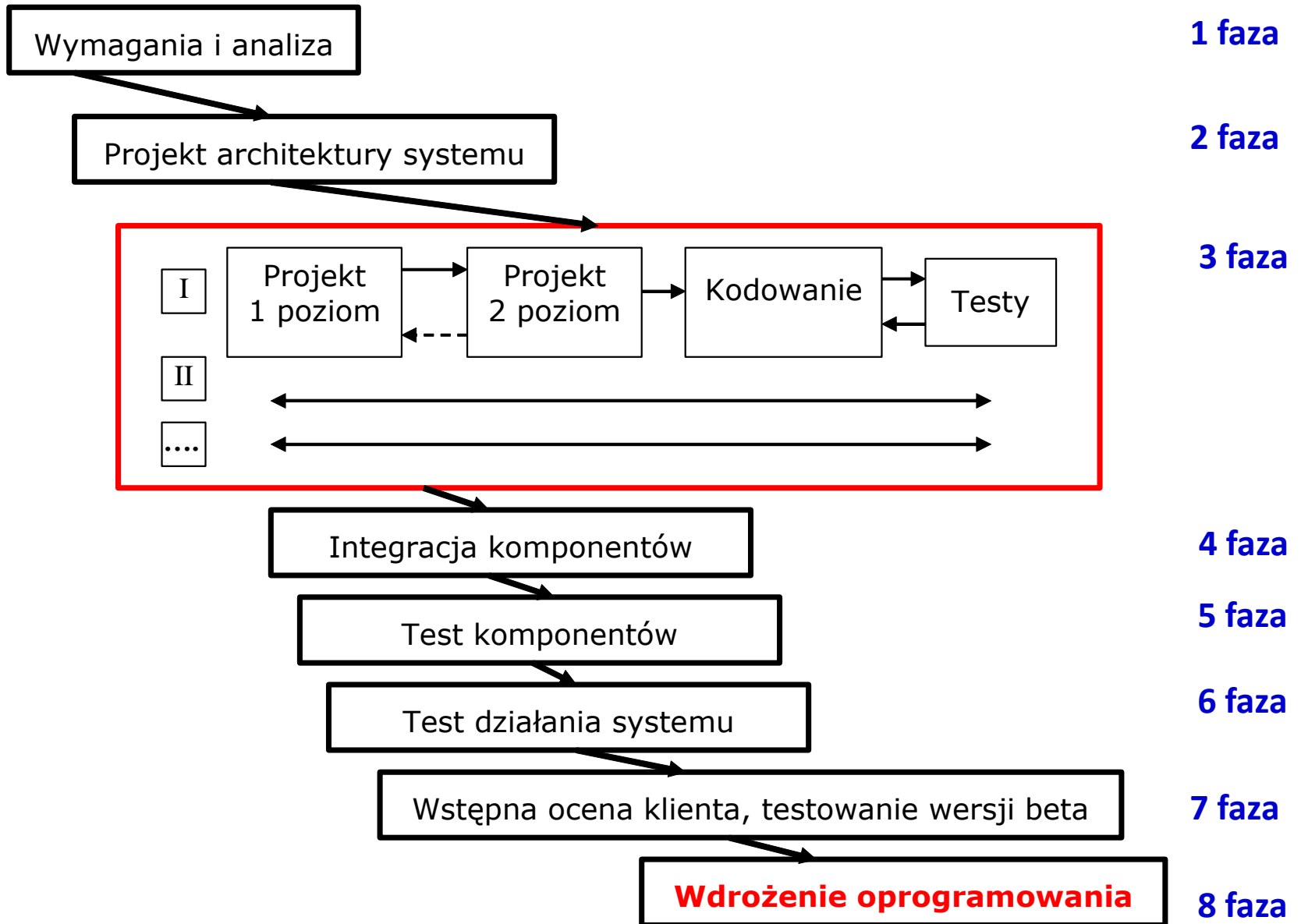
– Narzędzia: Bugfixes, Patches, Service Packs, New Releases

– Jak zaktualizować oprogramowanie dostępne dla klientów

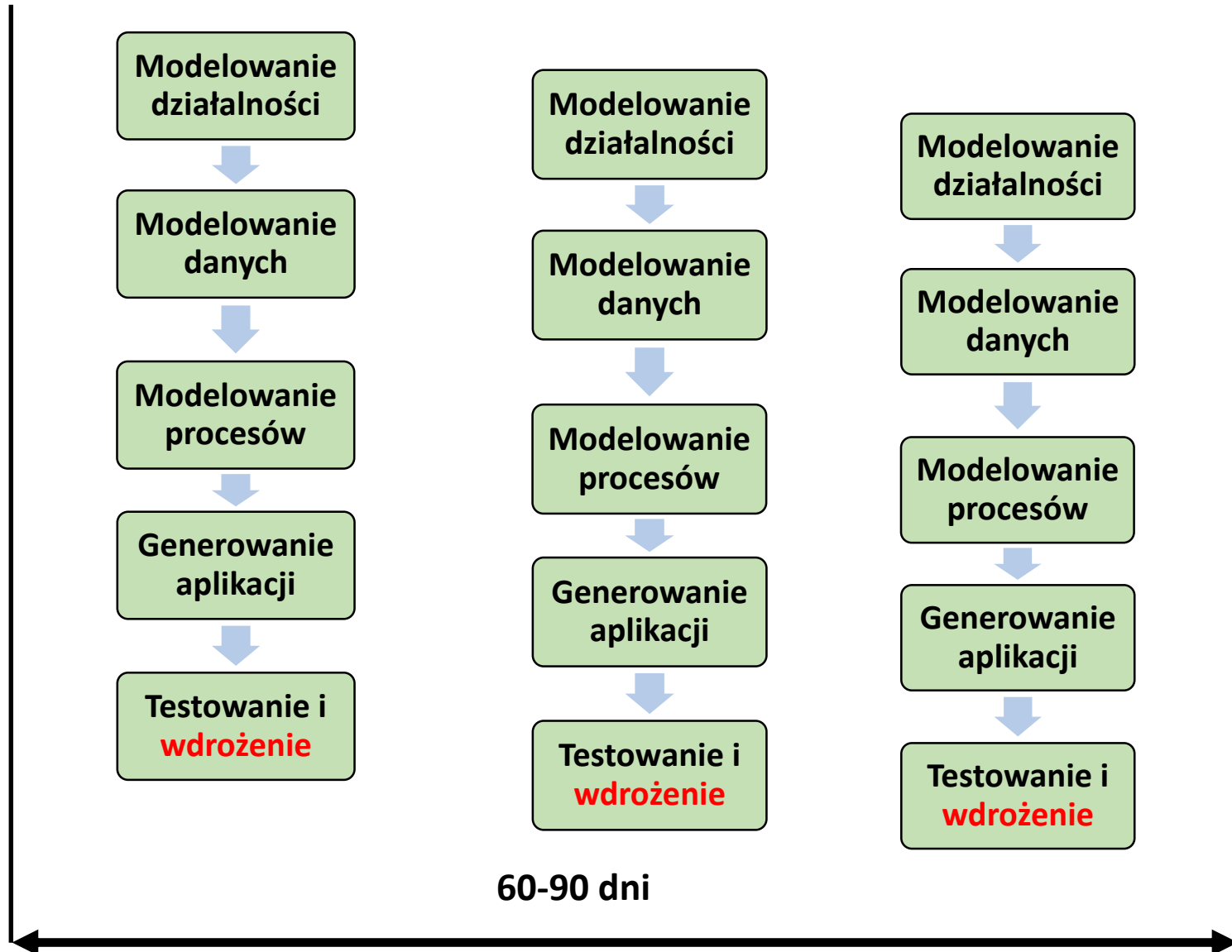
Narzędzia do wdrażania oprogramowania

- Narzędzia do wdrażania oprogramowania sprawiają, że proces dystrybucji oprogramowania i aktualizacji może być **wykonywany automatycznie lub w sposób zaplanowany**.
- Najlepsze narzędzia działają na różnych platformach i typach infrastruktury, **ułatwiając usprawnienie przepływu pracy w preferowanym środowisku**.
- Ciągła integracja i ciągłe wdrażanie podczas wytwarzania oprogramowania mogą być wykorzystywane do wdrażania aktualizacji oprogramowania dla użytkowników końcowych.
- Wybór najlepszego narzędzia do wdrażania oprogramowania powinien być dopasowany do procesu wytwarzania oprogramowania, w którym wdrażanie jest jednym z etapów
- **Przykłady narzędzi - Bamboo, TeamCity, Jenkins**

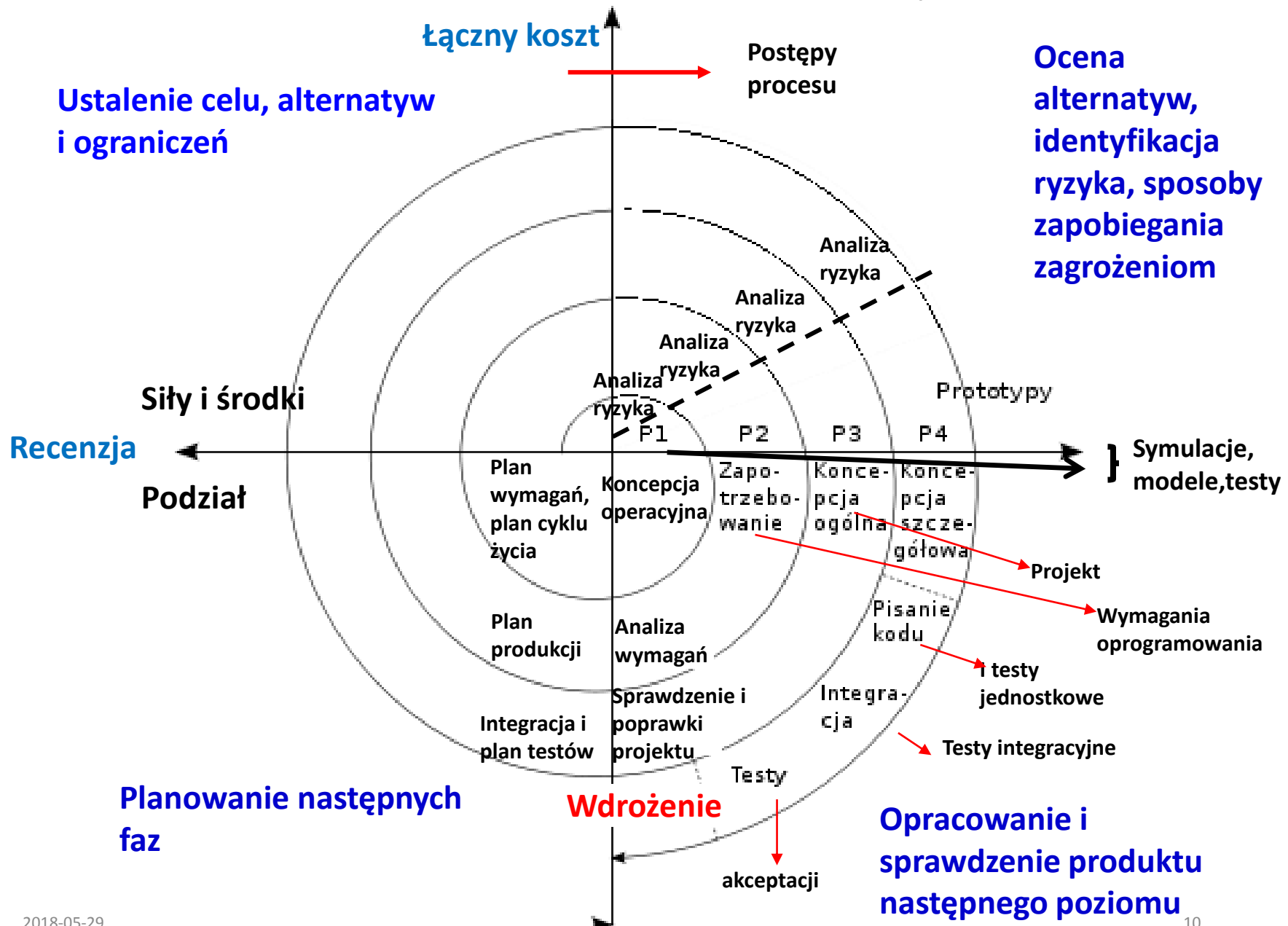
Proces kaskadowy (wykład 9)



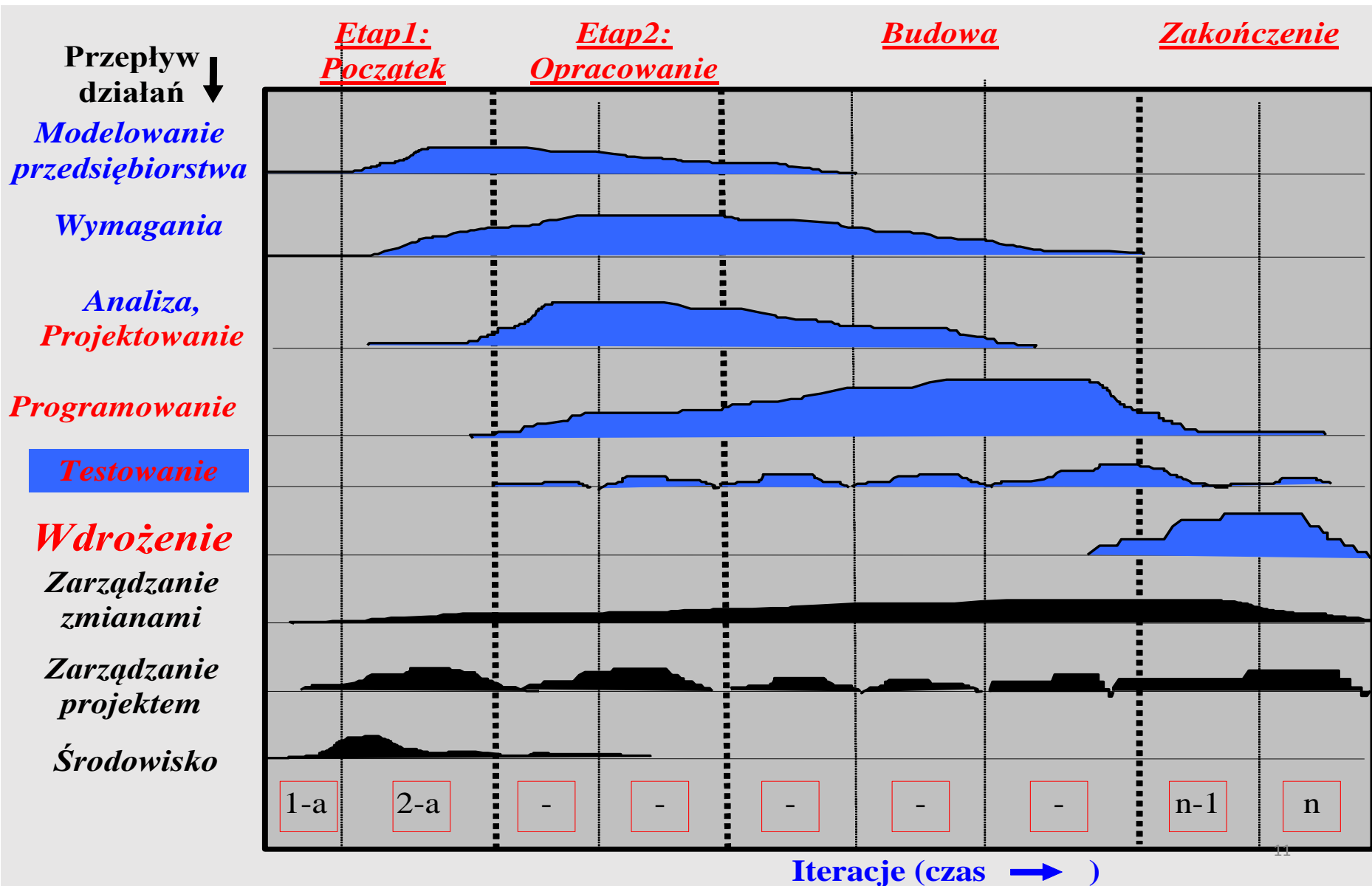
Proces RAD - szybka rozbudowa aplikacji (wykład 9)



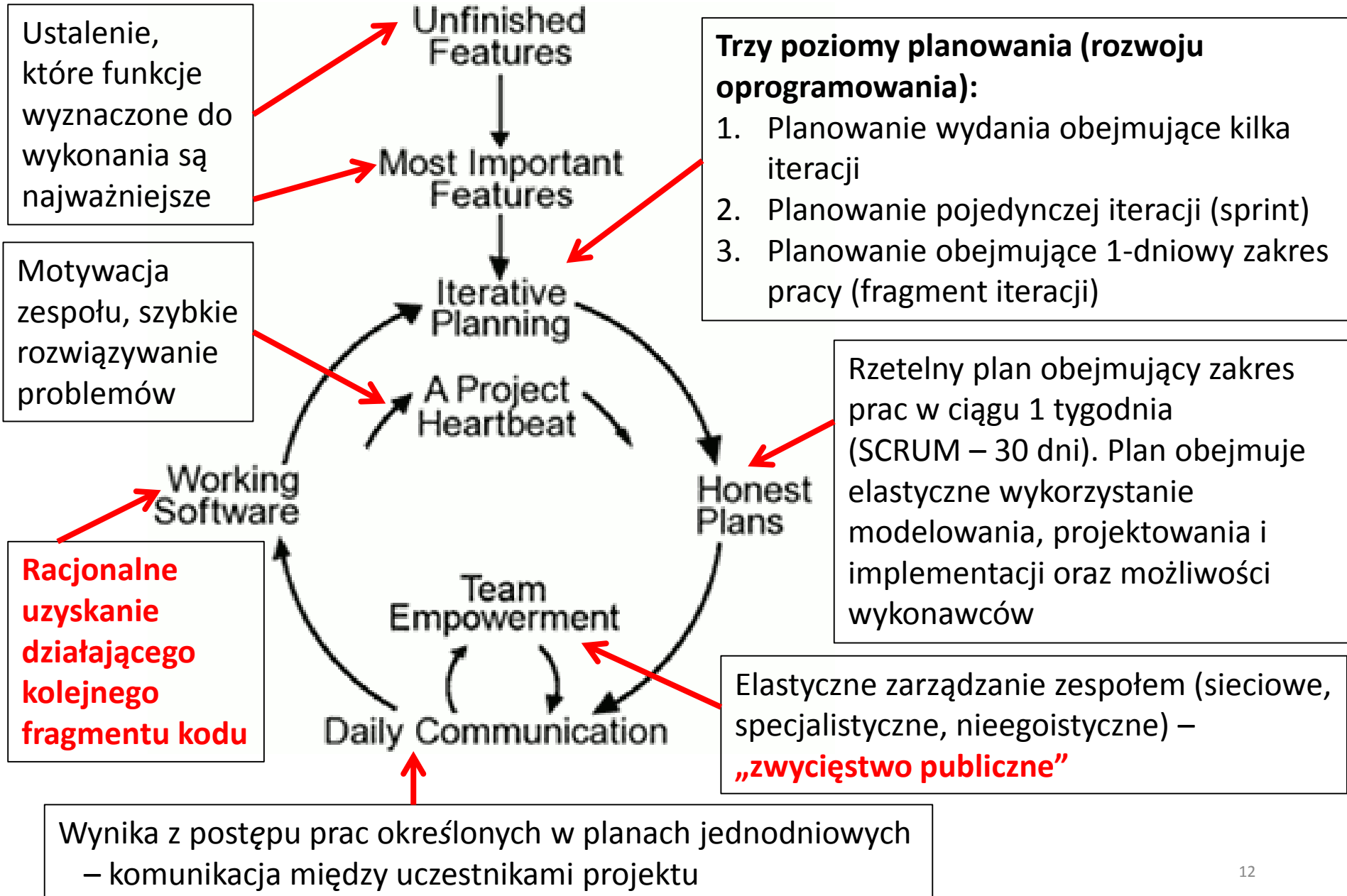
Proces ewolucyjny spiralny (wykład 9)



Ewolucyjny zunifikowany iteracyjno- przyrostowy proces tworzenia oprogramowania, (wykłady 1, 9)



Proces ewolucyjny zwinny XP (wykład 9)



Pielęgnacja oprogramowania

Pielęgnacja oprogramowania

[http://home.hit.no/~hansha/documents/software/software_development/topics/resources/Software Maintenance Overview.pdf](http://home.hit.no/~hansha/documents/software/software_development/topics/resources/Software_Maintenance_Overview.pdf)

Zanim gotowe oprogramowanie zostanie przekazane użytkownikowi, jego wersje są badane jako:

- Wersja alfa – wersja badana przez autorów oprogramowania
- Wersja beta – wersja robocza do testowania przez użytkownika
- RC (Release Candidate) – wersja testowa kandydująca przekazana użytkownikowi
- RTM (Release To Manufacturing) – wersja do wdrożenia, przekazana użytkownikowi

Konserwacja (po wdrożeniu oprogramowania)

- Patches - małe poprawki
- SP (Service Packs) - wiele małych poprawek i pakietów poprawek
- ...

Pielęgnacja oprogramowania – 4 kategorie

[http://home.hit.no/~hansha/documents/software/software_development/topics/resources/Software Maintenance Overview.pdf](http://home.hit.no/~hansha/documents/software/software_development/topics/resources/Software_Maintenance_Overview.pdf)

E. J. Braude i M. E. Bernstein, Inżynieria oprogramowania: Modern Approaches, 2 ed .: Wiley, 2011.

- **Naprawcza (Corrective)**

- Naprawa błędów wynikających z istniejących wymagań. Te usterki są zazwyczaj wykrywane przez klientów, gdy zaczynają używać oprogramowania

- **Adaptacyjna (Adaptive)**

- Dostosowanie oprogramowania do zmian w środowiska operacyjnego, np. gdy wydano nowy system operacyjny lub nową wersję sprzętu.

- **Perfektywna (Perfective)**

- Nowe funkcje wynikające z nowych potrzeb użytkowników.

- **Zapobiegawcza (Preventive)**

- Zmiany w oprogramowaniu, które ułatwiają konserwację.
- Zmiany wykonane w ramach 3 kategorii (Corrective, Adaptive i Perfective) sprawiają, że oprogramowanie jest bardziej złożone, trudniejsze w utrzymaniu. Konserwacja profilaktyczna w formie **Refaktoryzacji** powinna być wykonywana regularnie.

Software Maintenance

3 Categories (according to I. Sommerville, *Software Engineering*):

1. Fault Repairs

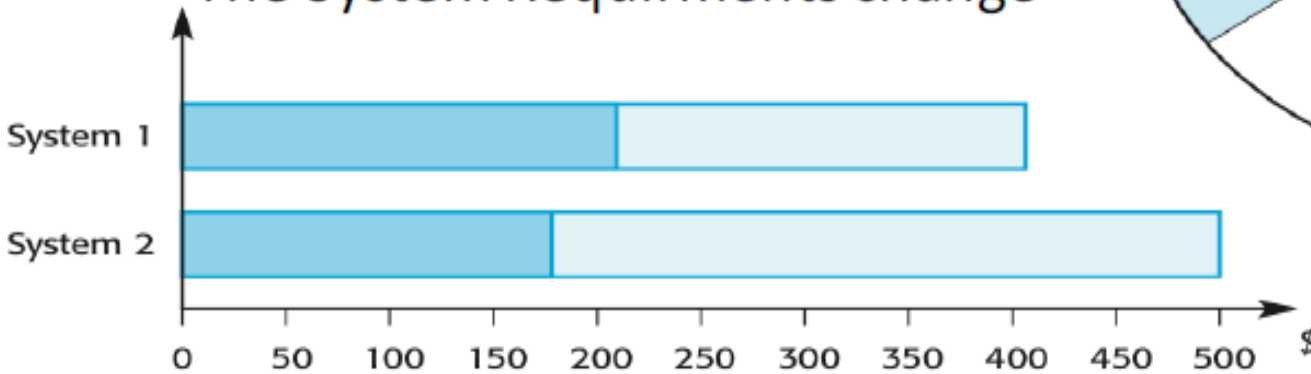
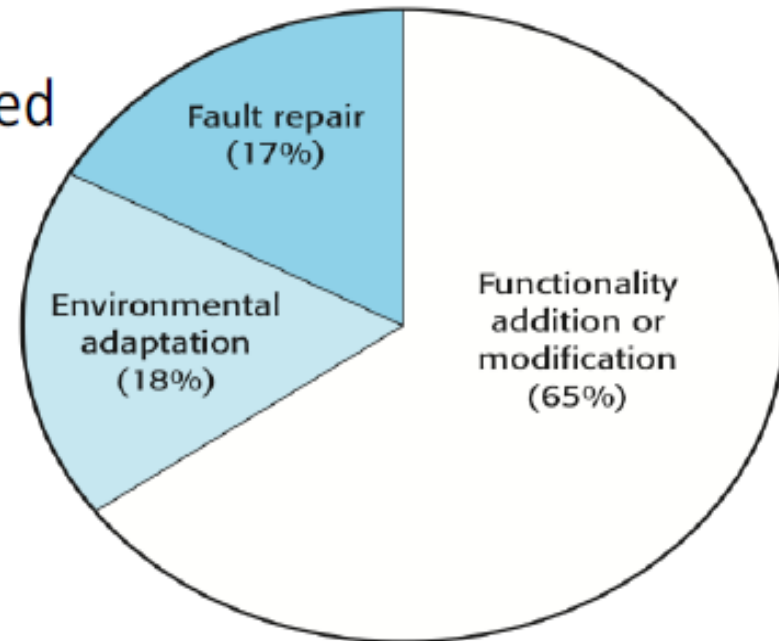
- Fixing Errors after Software is released

2. Environmental Adaption

- OS, Hardware, etc. changes

3. Functionality Addition

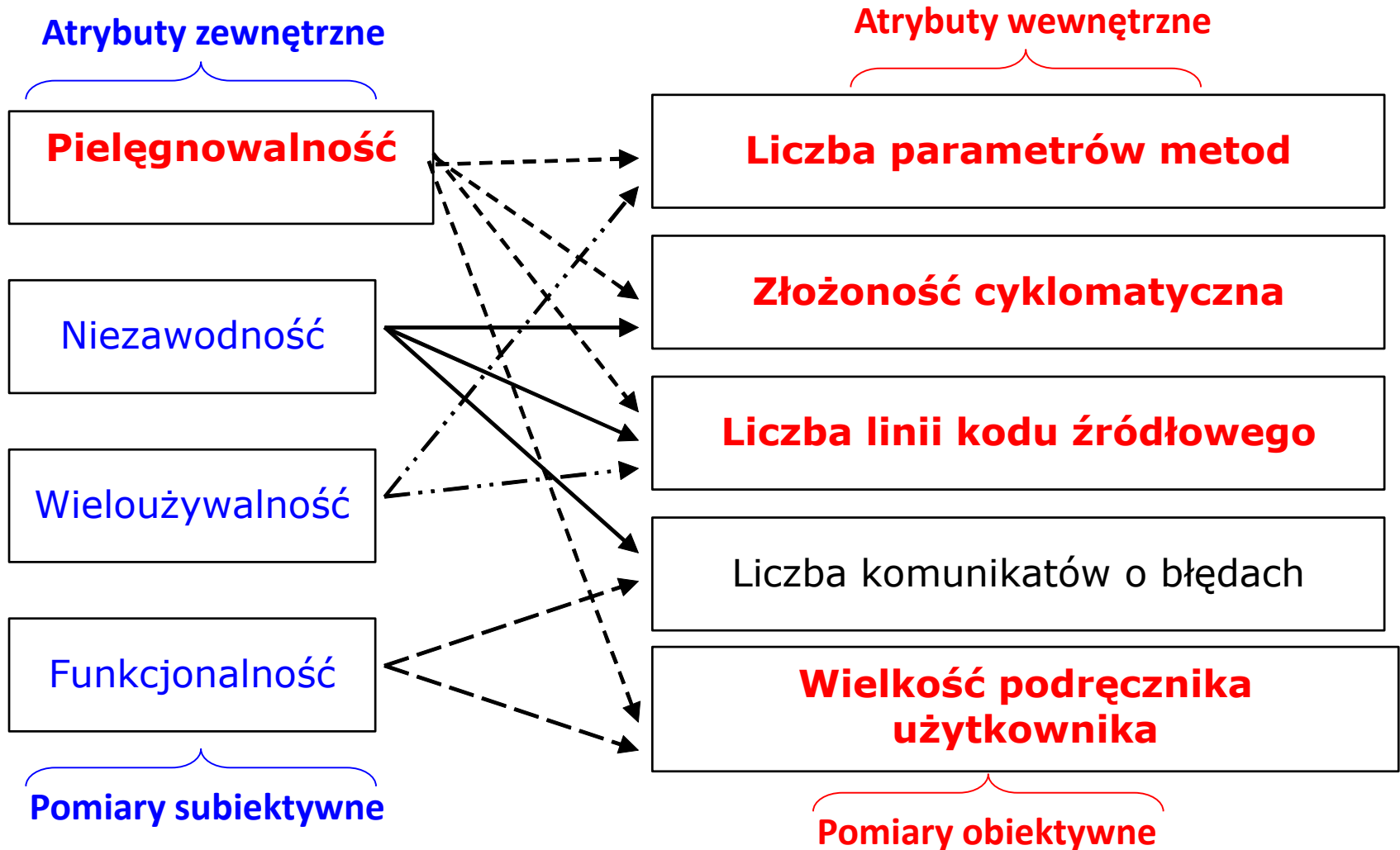
- The System Requirments change



■ Development costs ■ Maintenance costs

I. Sommerville, *Software Engineering*, 9 ed.: Pearson, 2010.

Przykłady powiązania metryk kodu z oceną **pielęgowalności** oprogramowania (wykład 12)

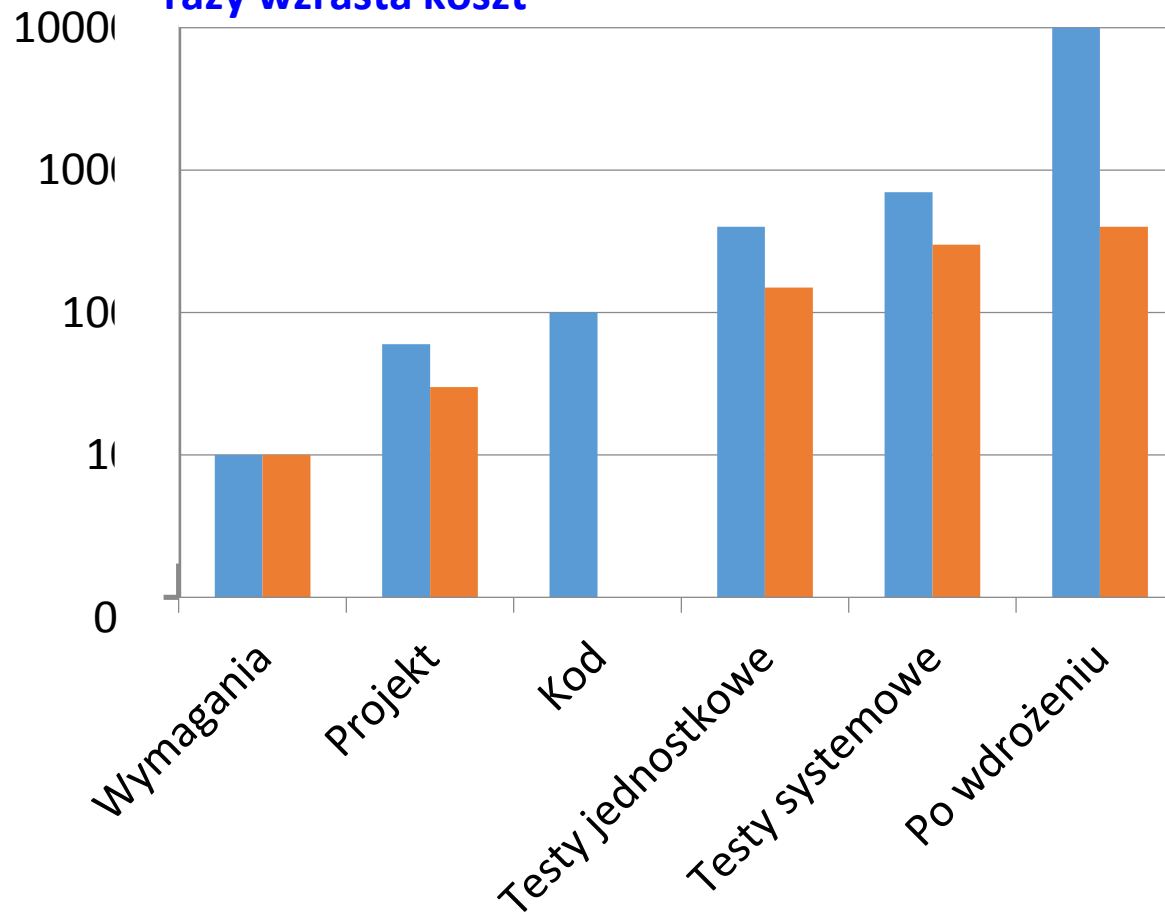


Przykłady powiązania metryk kodu z oceną pielęgnowalności oprogramowania (wykład 12)

Metryka	Jakość					Koszt	Funkcjonalność
	Stopień osiągniętej abstrakcji	Wieloużywalność	Zrozumiałość	Pielęgnowalność	Testowalność (niezawodność)		
LOC			+	+	+	+	
LCOM	+	+	+	+	+		
DIT			+	+	+		
NOC				+	+		
McCabe			+	+	+	+	
S/C		+	+	+	+	+	
CBO					+		
WMC					+		
RFC					+		

Względny koszt poprawiania błędów (wykład 10)

Względny koszt poprawiania błędu – ile razy wzrasta koszt



Przykład:

LOC = 200 000,
czas_usuwania_błędów
= 7053h,
koszt_h = 40 USD,
liczba_błędów = 3112,
cały_koszt = 282120 USD,
koszt_usuwania_błędu
≈ 91 USD,
a po wdrożeniu:
3640 ÷ 91000 USD

Usuwanie i wzmacnianie się błędów bez przeglądów technicznych (wykład 10)

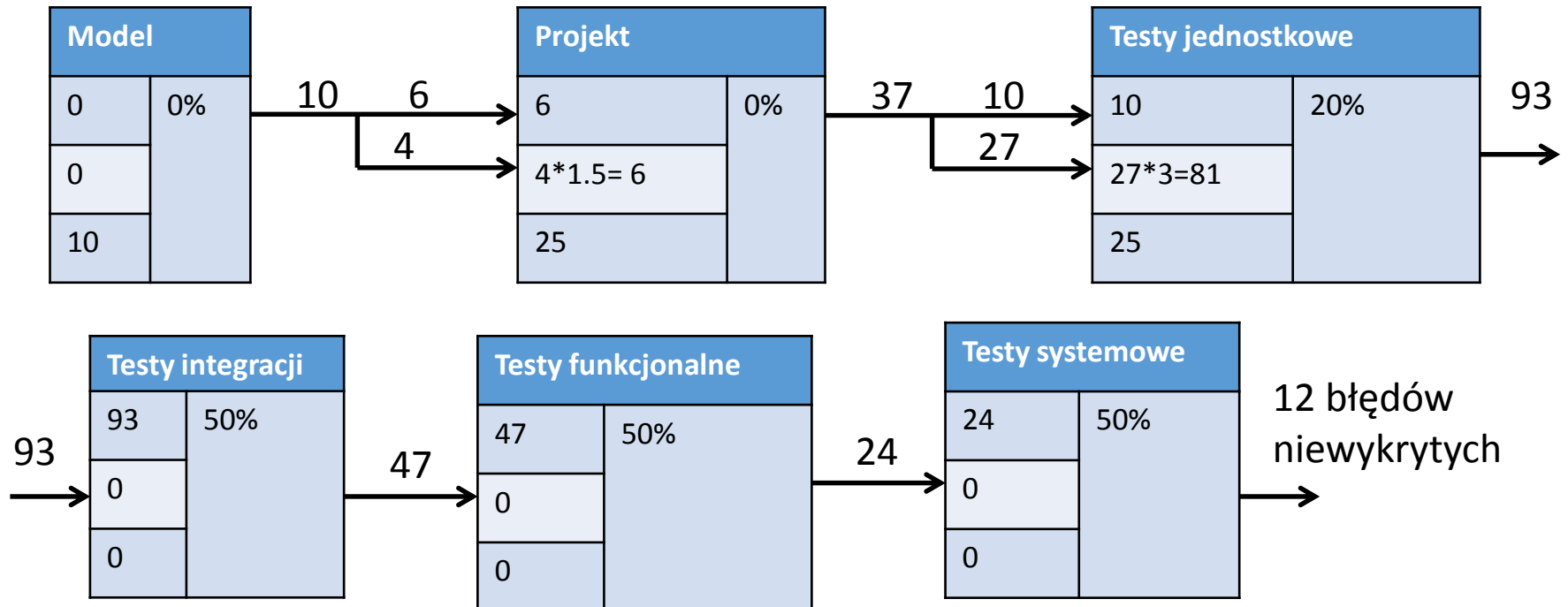
Błędy z poprzedniego etapu



Faza procesu	
Stare błędy	Procentowa skuteczność usuwania błędów
Wzmocnione błędy 1 : x	
Nowe błędy	



Błędy przekazane do następnego etapu



Usuwanie i wzmacnianie się błędów po wprowadzeniu przeglądów technicznych (wykład 10)

