



Politechnika Wroclawska

INFORMATYCZNE NARZĘDZIA ZARZĄDZANIE PROJEKTAMI

2019/2020

dr Sebastian Tomczak



Metody tradycyjne i nowoczesne w zarządzaniu projektami IT



AGENDA

- Podejście kaskadowe
- Podejście spiralne
- Podejście przyrostowe
- Podejście iteracyjne
- Podejście zwinne



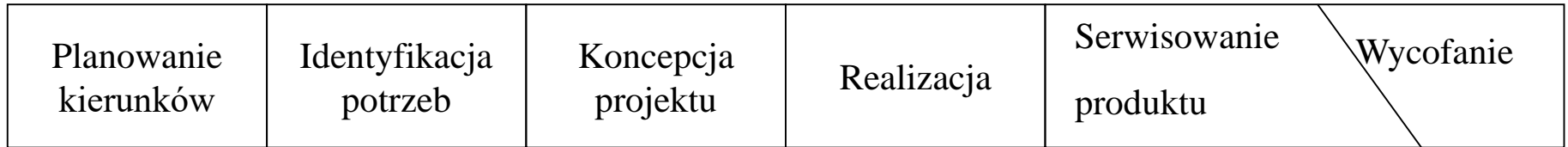
CYKL ŻYCIA PROJEKTU I PRODUKTU

- Cykl życia projektu – zbiór etapów projektu przeprowadzanych w określonym porządku, od pomysłu na projekt aż do jego zakończenia.
- Cykl życia produktu – zbiór etapów produktu przeprowadzanych w określonym porządku, od pomysłu na produkt do jego wykonania wraz z jego eksploatacją aż do zaniechania.

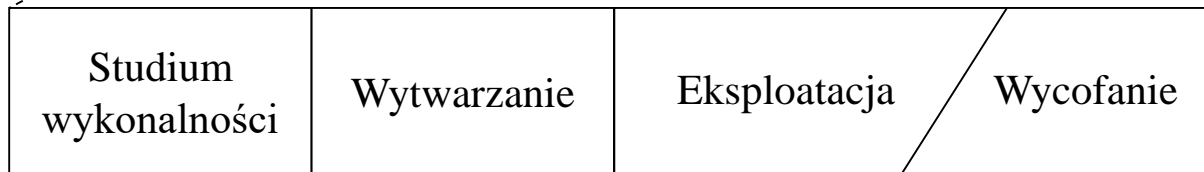


RELACJE CYKLI ŻYCIA

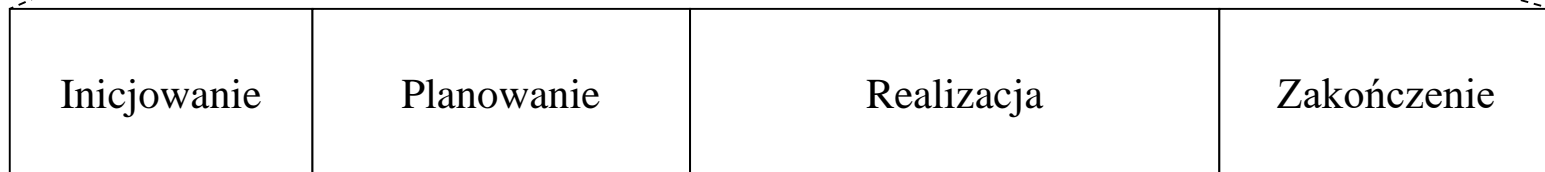
Cykl życia organizacji



Cykl życia produktu



Cykl życia projektu



CYKL ŻYCIA PROJEKTU TWORZENIA OPROGRAMOWANIA

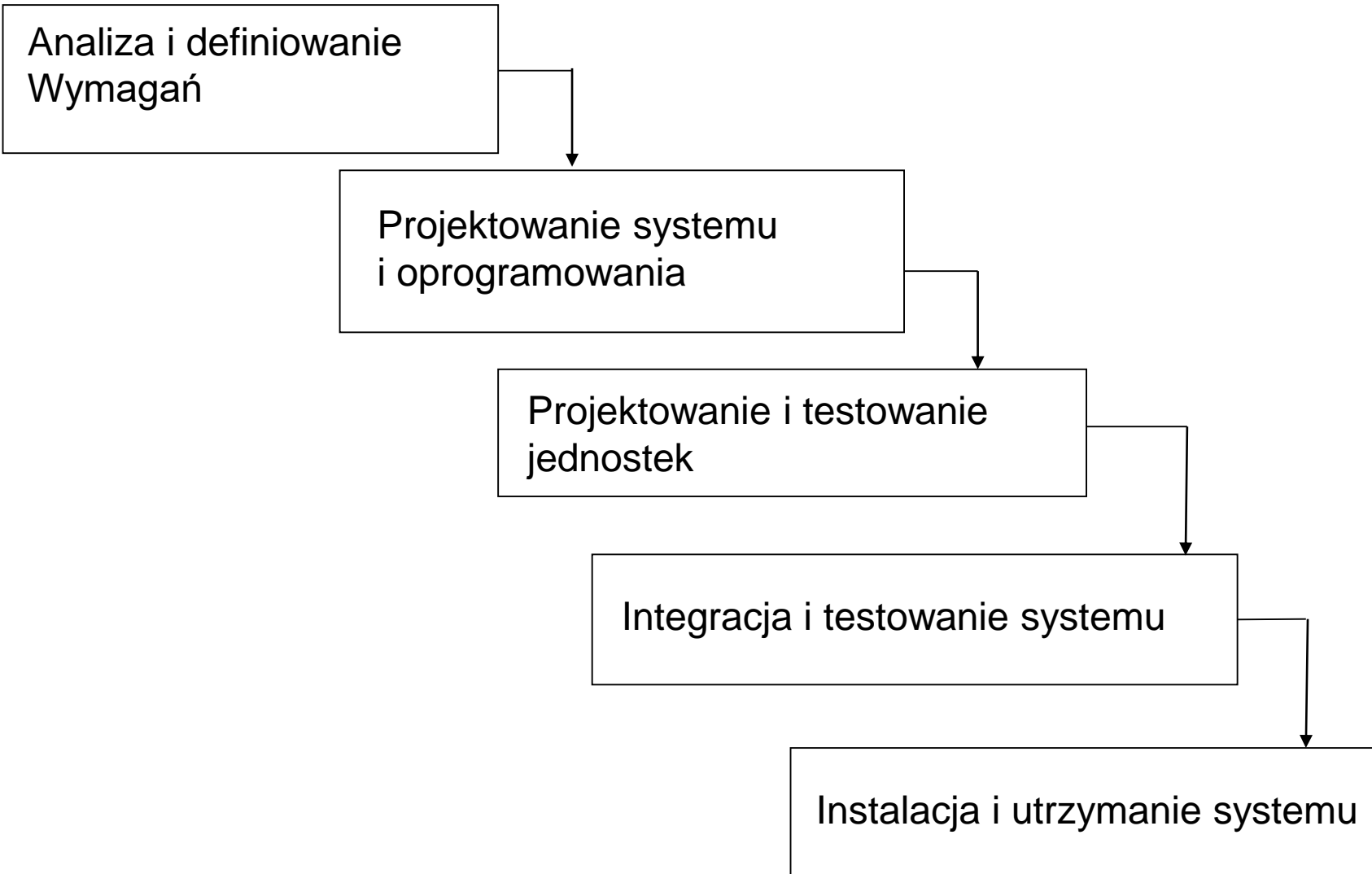




PODEJŚCIE KASKADOWE

- Kładzie silny nacisk na wstępne planowanie i kontrolę z bardzo ograniczoną elastycznością.
- Podejście to można zakwalifikować jako „sterowane planem” lub „przewidywalne”, gdyż wykorzystują wstępne planowanie, aby przewidzieć koszt i ustalić harmonogram projektu.
- Kładzie zwykle nacisk na kontrolowanie zmian podczas trwania projektu, aby zarządzać zakresem pracy i zagwarantować, że początkowe przewidywania i harmonogramy pozostaną aktualne.

MODEL KASKADOWY





KONSEKWENCJE PODEJŚCIA KASKADOWEGO

- brak możliwości zebrania informacji zwrotnej na temat naszego produktu,
- brak możliwości otrzymania produktu wcześniej niż planowano,
- możliwość zarabiania na produkcie dostępna dopiero po wdrożeniu,
- wysokie ryzyko techniczne,
- wysokie ryzyka biznesowe.



PODEJŚCIE SPIRALNE

- Model ten łączy cechy modelu kaskadowego oraz prototypowego.
- Proces ten ma postać spirali, w której każda pętla przedstawia kolejne fazy procesu.
- Każda faza składa się z czterech etapów:
 1. Ustalenie celów
 2. Rozpoznanie i redukcja zagrożeń
 3. Tworzenie i testowanie
 4. Planowanie następnej iteracji.

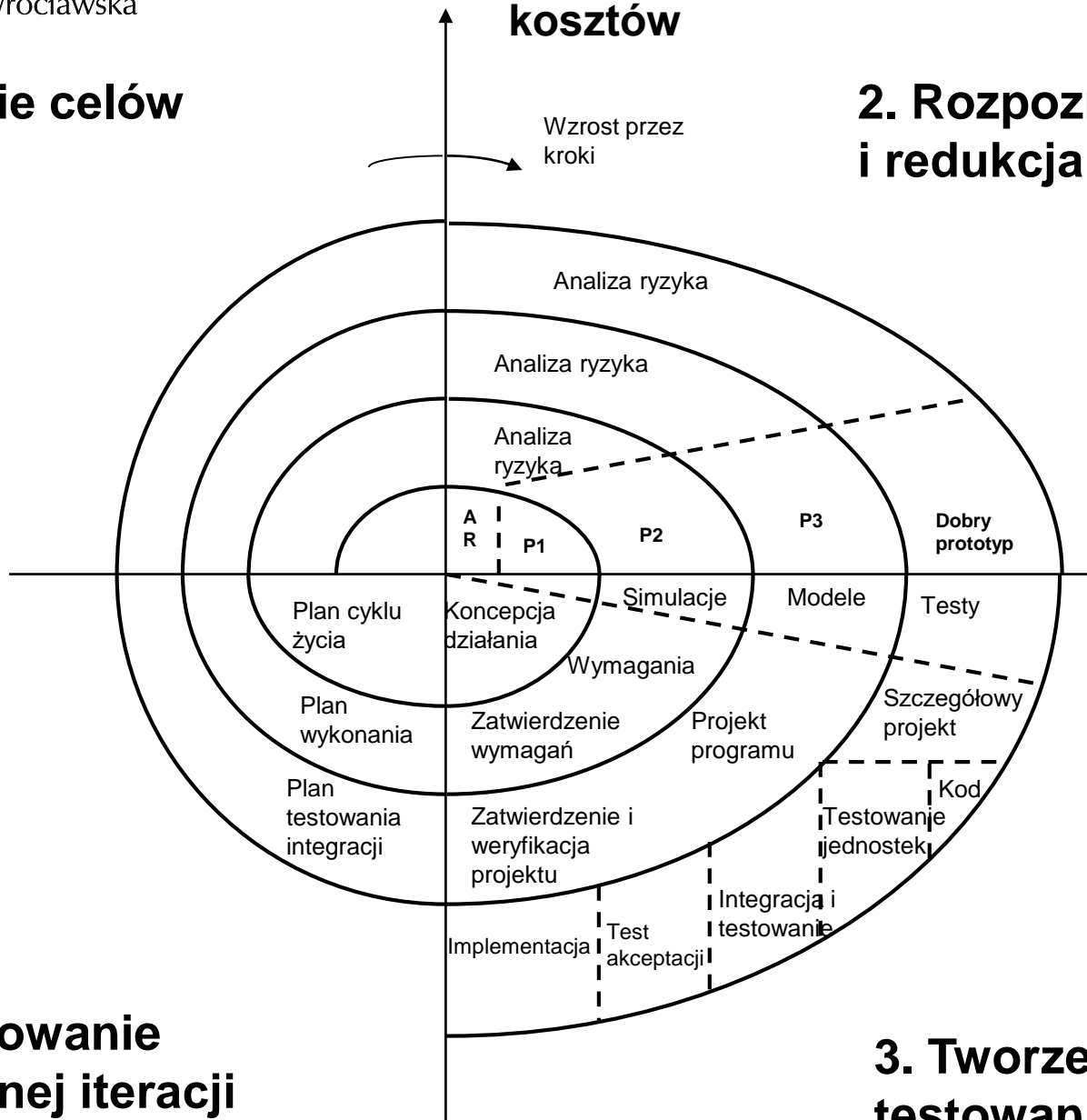


Kumulacja kosztów

1. Ustalenie celów

2. Rozpoznanie i redukcja zagrożeń

Przegląd części



P1,2,3 - prototypy

4. Planowanie następnej iteracji

3. Tworzenie i testowanie



KONSEKWENCJE PODEJŚCIA SPIRALNEGO

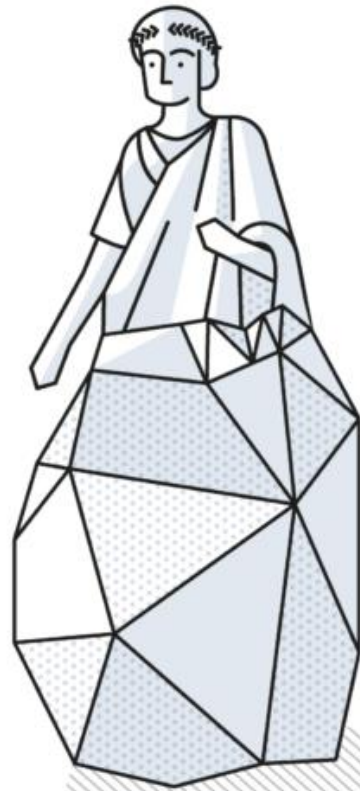
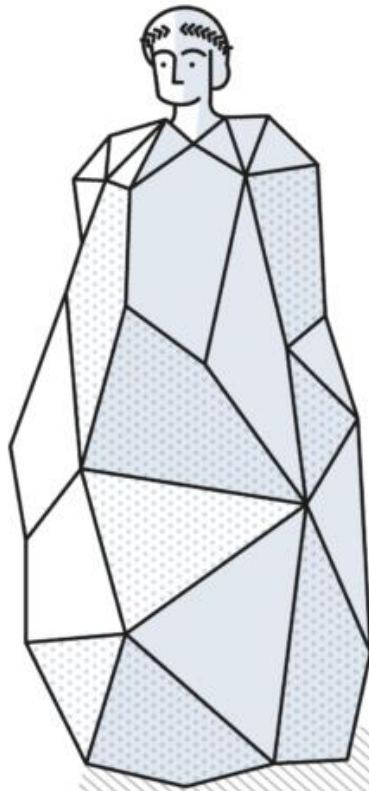
- Model ewolucyjny, powtarzalność oparta na prototypowaniu
- Zastosowanie - do dużych projektów
- Każde okrążenie dotyczy jednego elementu produktu: koncepcja, wymagania, projekt, kod
- Umożliwia zmiany w rozwoju produktu – zarządzanie zmianami
- Konieczność zarządzania ryzykiem
- Wczesna eliminacja błędów
- Powtórne wykorzystanie wcześniej wykonanych części
- Każdy cykl zakończony przeglądem wykonanym przez kluczowych członków zespołu
- Wielokrotne powtarzanie ekspertyz analizy ryzyka
- Wymaga dużej wiedzy i doświadczenia od kierownika procesu
- Trudności w opracowaniu i kontroli kontraktu



PODEJŚCIE PRZYROSTOWE

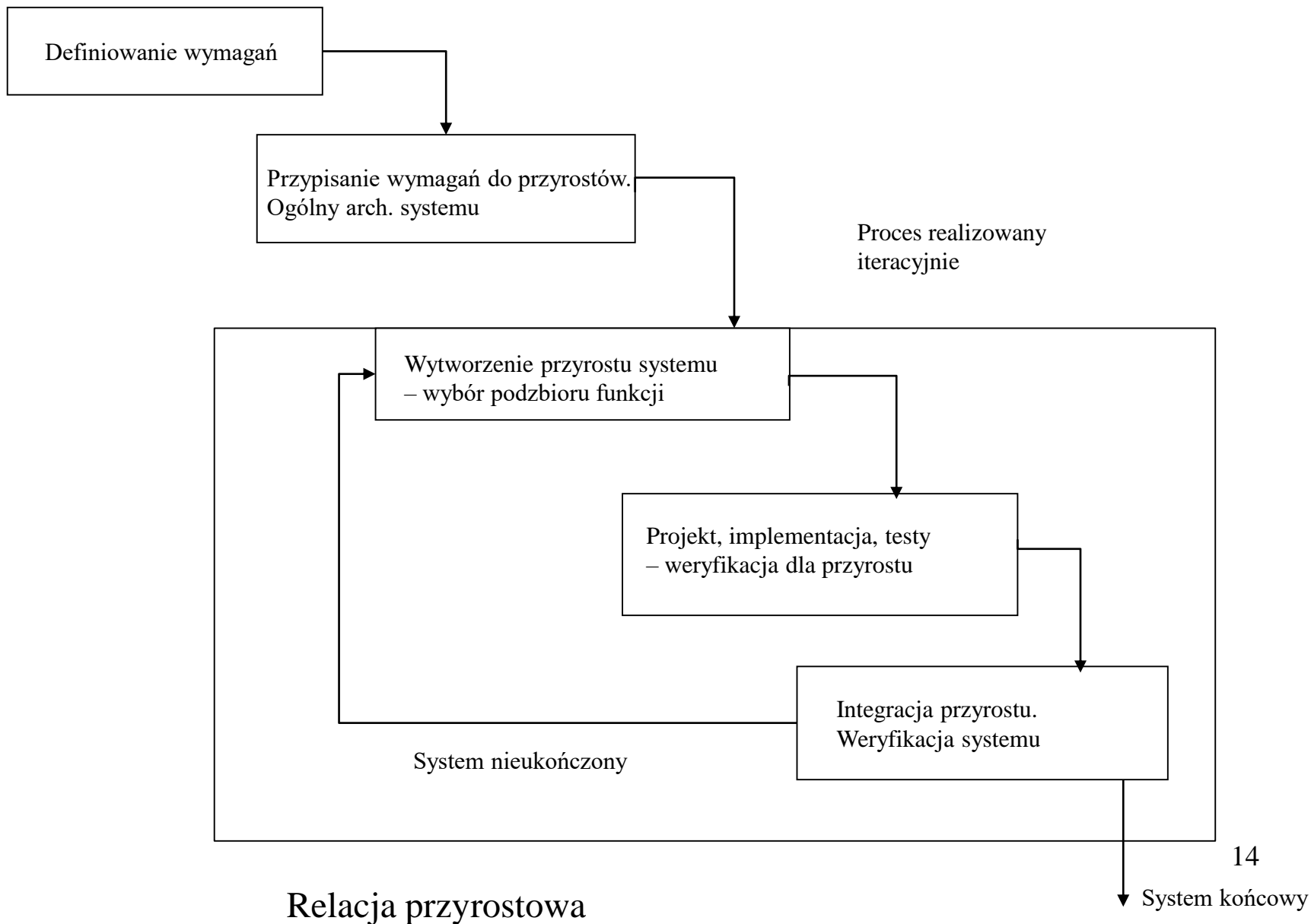
- Wymaga posiadania jasnej wizji produktu, której realizacja prowadzona jest w powtarzalnych, krótkich krokach o stałej długości, trwających zwykle 1 lub 2 tygodni.
- Każdy krok składa się z małego kawałka analizy, projektowania, właściwej implementacji oraz testów

PODEJŚCIE PRZYROSTOWE





TWORZENIE PRZYROSTOWE





KONSEKWENCJE PODEJŚCIA PRZYROSTOWEGO

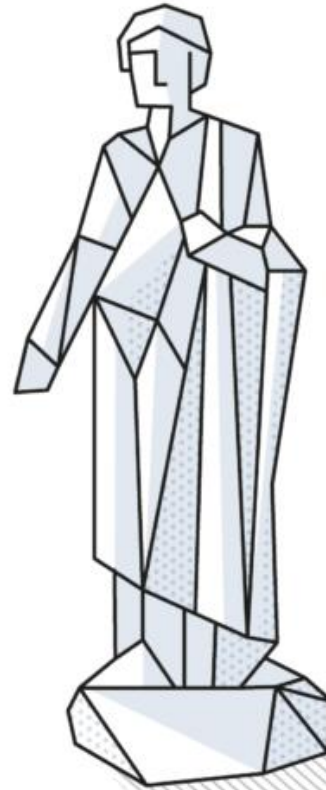
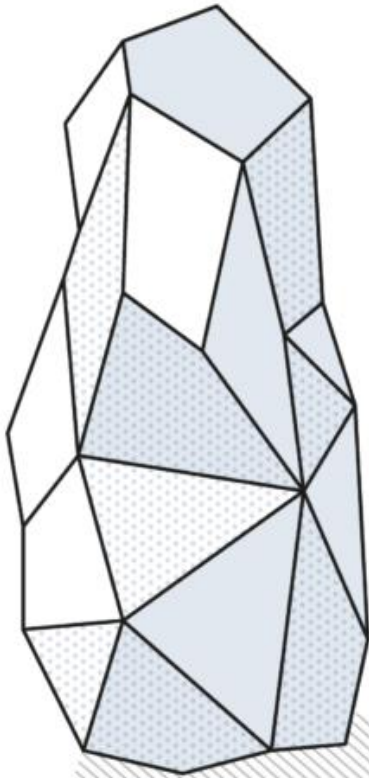
- częściowa możliwość zebrania informacji zwrotnej na temat naszego produktu,
- częściowa możliwość otrzymania produktu wcześniej niż planowano,
- częściowa możliwość wydania mniej pieniędzy, niż planowano,
- częściowa możliwość obniżenia ryzyk projektowych.



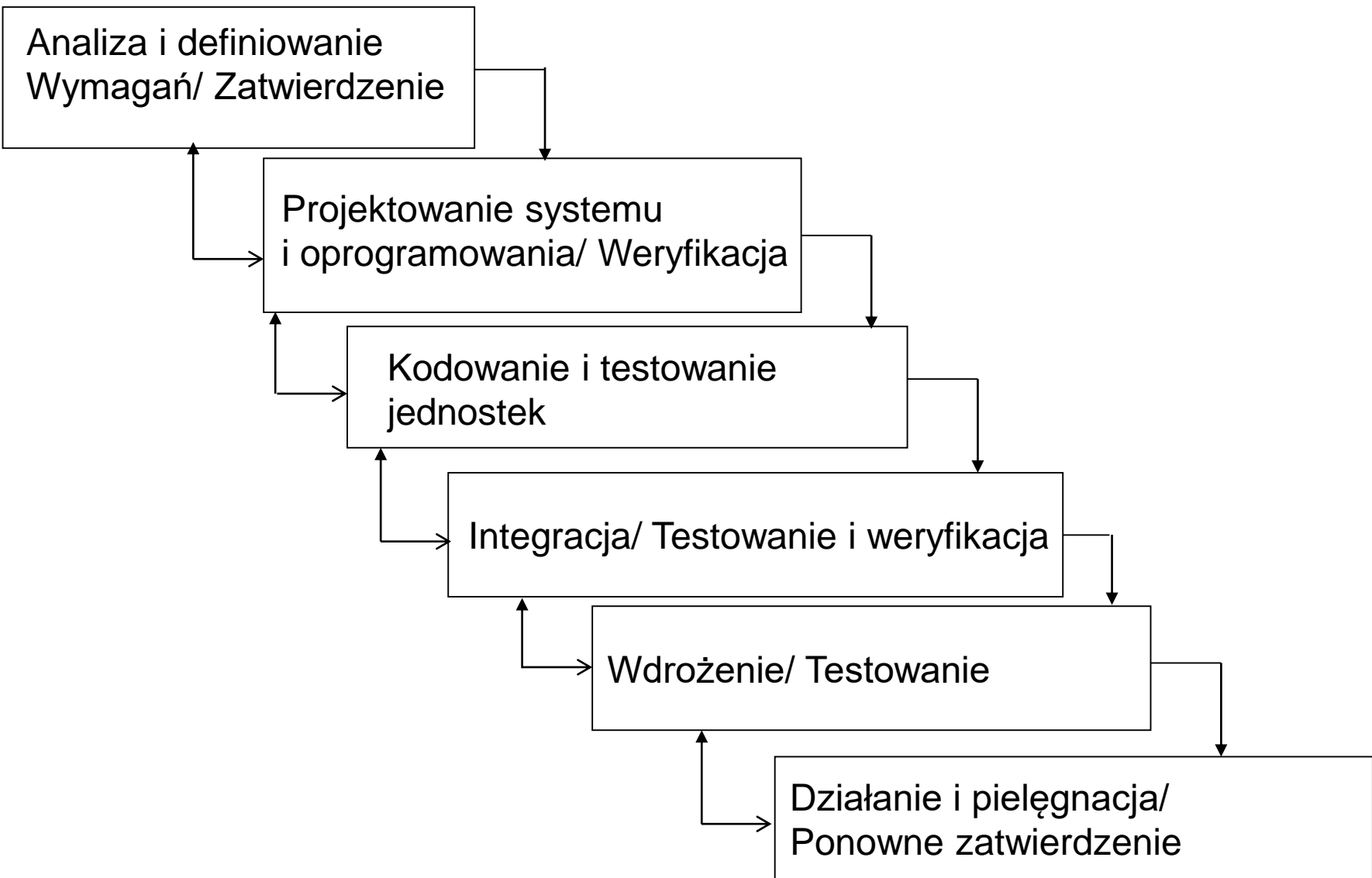
PODEJŚCIE ITERACYJNE

- Wymagania i projekt są modyfikowane poprzez serie iteracji prowadzących do otrzymania systemu satysfakcjonującego rozwijające się potrzeby klienta
- Sesje „sprzężenia zwrotnego” i zasada wzajemnego uczenia się
- Zwiększenie zrozumienia definicji wymagań
- Łatwiejsze zarządzanie zmianami
- Umożliwienie rozpoczęcia tworzenia aplikacji dla podzbioru wymagań - analiza dla każdego produktu częściowego
- Wczesna neutralizacja zagrożeń
- Zwiększenie możliwości ponownego użycia kodu
- Łatwiejsze dostosowanie końcowego produktu do zmieniających się wymagań

PODEJŚCIE ITERACYJNE



ZMODYFIKOWANY MODEL KASKADOWY





KONSEKWENCJE PODEJŚCIA ITERACYJNEGO

- możliwość zebrania informacji zwrotnej na temat naszego produktu,
- możliwość otrzymania produktu wcześniej niż planowano,
- możliwość niewykorzystania całego budżetu,
- wczesne obniżenie ryzyk projektowych.



NARZĘDZIA – PODEJŚCIE TRADYCYJNE

Primavera P6 Professional Project Management

ORACLE®

PRIMAVERA

MS Project





NOWOCZESNE METODY W ZARZĄDZANIU PROJEKTAMI IT



ADAPTACYJNE ZARZĄDZANIE PROJEKTAMI

- Adaptacyjne zarządzanie projektami - zbiór różnych metodyk, określanych jako zwinne bądź lekkie, oraz narzędzi stosowanych w zarządzaniu złożonymi i innowacyjnymi projektami - głównie informatycznymi.
- Powstanie nowego podejścia było reakcją na mało elastyczne metody zarządzania projektami informatycznymi, uważane za zbyt sformalizowane i mało efektywne.
- Dokument „Manifesto for Agile Software Development” (2001) zainicjował głębokie przemiany w środowiskach programistycznych, przyjętych też w innych obszarach zarządzania projektami.



TREŚĆ MANIFESTU AGILE

- *Poprzez wytwarzanie oprogramowania oraz pomaganie innym w tym zakresie odkrywamy lepsze sposoby realizowania tej pracy.*
- *W wyniku tych doświadczeń zaczęliśmy przedkładać:*
 - *Ludzi i ich wzajemne interakcje(współdziałanie) ponad procedury i narzędzia.*
 - *Działające oprogramowanie nad wyczerpującą dokumentacją.*
 - *Współpracę z klientem nad negocjację umów.*
 - *Reagowanie na zmiany nad realizowanie planu.*
- *Oznacza to, że co prawdzie doceniamy to co wymieniono po prawej stronie, to jednak bardziej cenimy to co wymieniono po lewej*



AGILE (WG DR. DAVIDA RICO)

- Zdolność do tworzenia zmian i reagowania na nie, aby odnieść korzyści w burzliwym globalnym środowisku biznesowym.
- Zdolność do szybkiej zmiany priorytetów wykorzystania zasobów, gdy zmieniają się wymagania, technologie lub wiedza.
- Szybka reakcja na nagłe zmiany rynkowe oraz pojawiające się zagrożenia dzięki intensywnej współpracy z klientem.
- Wykorzystanie ewolucyjnych, przyrostowych i iteracyjnych metod dostarczania produktu, aby dojść do rozwiązania optymalnego z punktu widzenia klienta.
- Maksymalizacja wartości biznesowej za pomocą procesów i dokumentacji realizowanych w odpowiedniej wielkości, tyle ile potrzeba oraz dokładnie na czas.



ZWINNOŚĆ I ADAPTACYJNOŚĆ

Punkty krytyczne „Agile” zespołu w projekcie

1. Od dwóch do ośmiu osób w jednym pokoju
2. Eksperti klienta na miejscu
3. Jednomiesięczne przyrosty
4. W pełni zautomatyzowane testy regresyjne
5. Doświadczeni programiści

Adaptacyjność - potrzeba analizy przeszłych działań



CECHY CHARAKTERYSTYCZNE METODYK AGILE

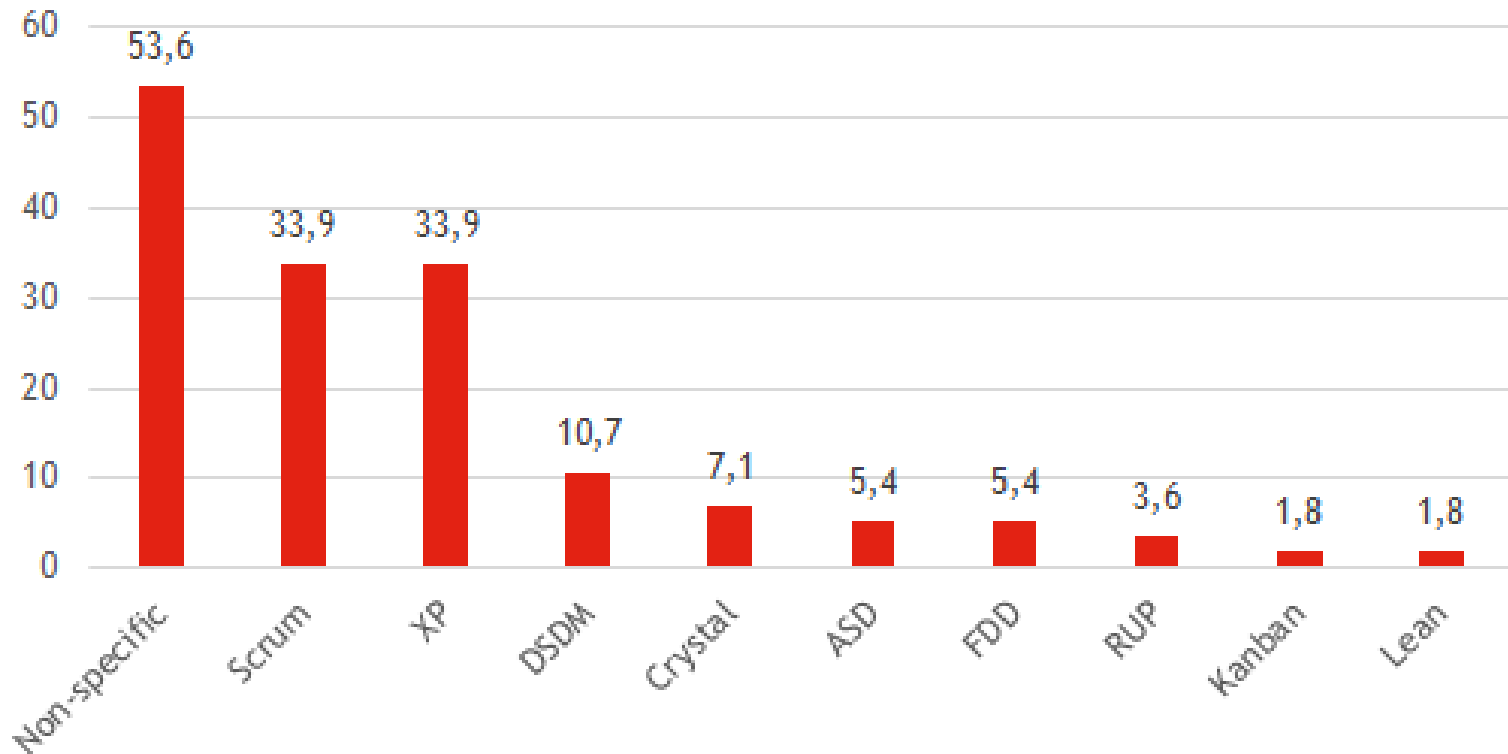
- Cechy
 - ograniczenie liczby dokumentów
 - planowanie iteracyjne, w krótszej perspektywie czasowej
 - ciągła współpraca z klientem
 - otwartość na zmiany
 - niewielkie zespoły projektowe
 - brak wydzielania faz projektowych
 - zdobywanie wymagań za pomocą wykonywalnego kodu
- Zalety
 - szybkie wydanie działającego produktu, każda budowa trwa od 2-10 tygodni
 - wykonawca i użytkownik bezpośrednio uświadamiają sobie wymagania (kompletność i spójność)
 - w dowolnym punkcie procesu tworzenia klient posiada działającą wersję produktu (z ograniczonymi możliwościami)
- Nieodpowiednie dla
 - zespołów przekraczających 50 osób
 - produktów wykonujących funkcje krytycznego bezpieczeństwa
 - kontraktów o ustalonym zakresie



PRZYKŁADY METODYK AGILE

- Extreme Programing (XP)
- Scrum, Kanban, Crystal, Lean Software Development (LSD)
- Dynamic Systems Development Method (DSDM)
- Adaptive Software Development (ASD)
- Feature Driven Development (FDD)

NAJCZEŚCIEJ STOSOWANE METODY AGILE





METODA SCRUM

- Literatura:

1. Cockburn A., *Agile Software Development. Gra zespołowa*, Helion 2008
2. Schwaber K., *Sprawne zarządzanie projektami metodą Scrum*, Microsoft Press 2005.
3. Schwaber K., Sutherland J., *SCRUM guide – Przewodnik po metodyce SCRUM*, tłum. A. Gajewska, T. Włodarek, 2010

- Podstawowe założenia:

- iteracje – najczęściej 30-dniowe przedziały czasowe - sprint
- przyrostowe tworzenie oprogramowania
- empiryczna kontrola procesu: przejrzystość, kontrola, adaptacja
- samorganizujący się zespół projektowy

- Narzędzia:

- **ATLASSIAN JIRA**



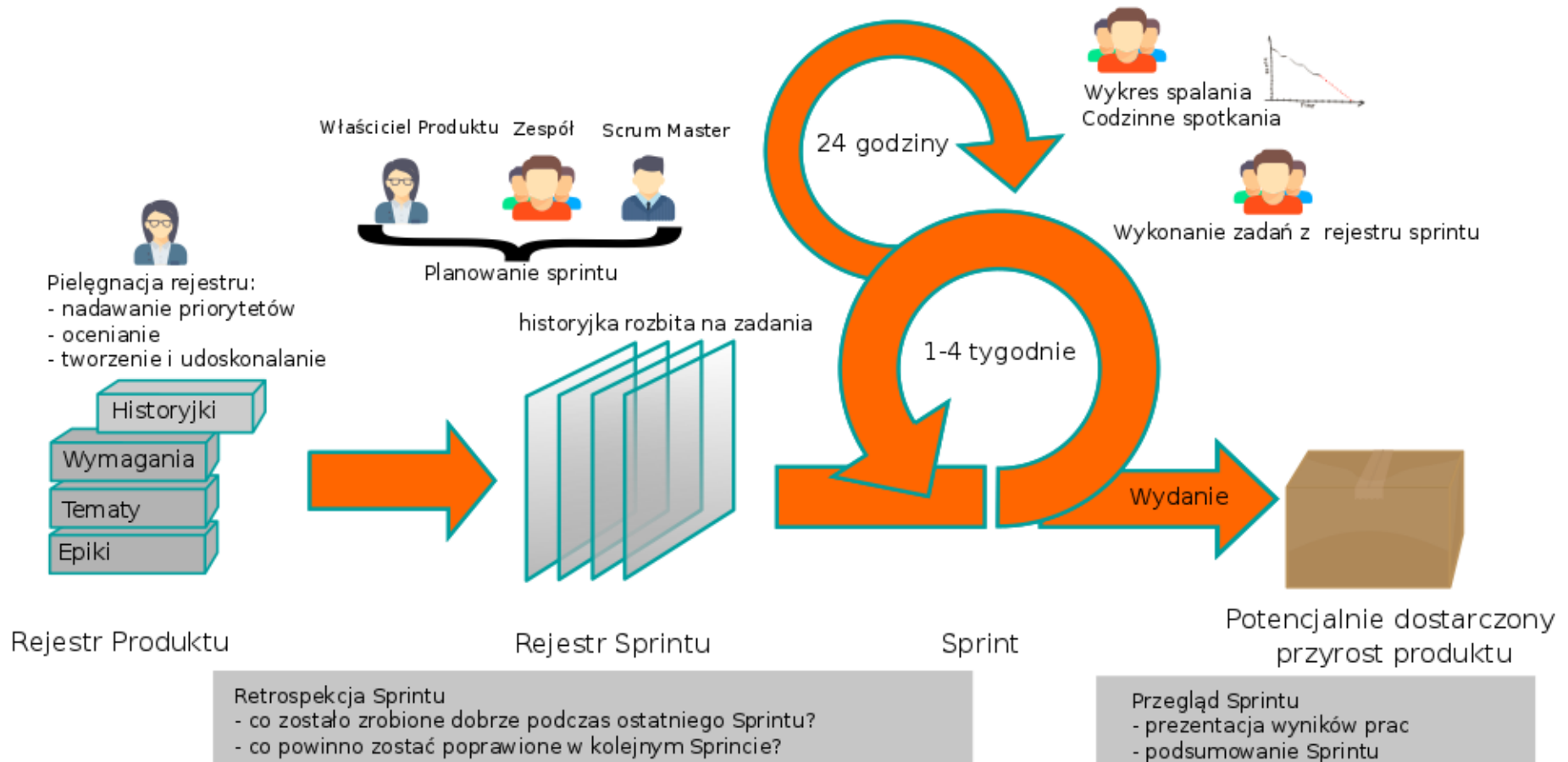


SCRUM

- jest elastycznym (*agile*) modelem tworzenia oprogramowania opartym na wielu małych zespołach pracujących w intensywny i współzależny sposób.
- obejmuje procesy podejmowania decyzji w czasie rzeczywistym na podstawie zachodzących zdarzeń i otrzymywanych informacji.



ILUSTRACJA PRZEBIEGU PROJEKTU WG SCRUM





SCRUM JAKO „SZKIELET PROCESU”

- Metoda Scrum jako zbiór praktyk i ról
- Główne role:
 - Mistrz (Scrum Master),
 - Właściciel produktu (Product Owner),
 - Zespół (Team)
- **Właściciel produktu**
 - Jest właścicielem definicji sukcesu, reprezentuje organizację i interesariuszy projektu
 - Kieruje produktem od sprintu do sprintu, aby zapewnić największy zwrot z inwestycji i dostarczyć pewną wartość dla organizacji.
 - Zarządza ROI przez określenie priorytetów oraz publikację planów. Jest jedynym właścicielem zaległości produktu. Określa plan rozwoju projektu poprzez wyznaczanie priorytetów zaległości.
 - Eliminuje błędy występowania wielu szefów, różnych opinii i zakłóceń.



ROLE W SCRUM

- **Mistrz**
- Jedna osoba z zespołu wciela się w rolę mistrza - ułatwia codzienną pracę zespołu, nie robi nic innego. Całe jego obciążenie to pełnienie roli Mistrza w pełnym wymiarze czasu pracy.
- Mistrz jest odpowiedzialny za zapewnienie, że zespół żyje wartościami i pracuje zgodnie z regułami metody Scrum.
- Jest tarczą zespołu projektowego dla agresywnych klientów, upewniając się, że zespół nie wychodzi ponad zobowiązania aktualnego sprintu.
- Mistrz staje się odpowiedzialny za usuwanie wszelkich przeszkód, które zgłaszane są przez zespół podczas spotkań.
- Rola mistrza jest zwykle pełniona przez kierownika projektu lub kierownika zespołu technicznego, ale może to być każda osoba z zespołu zarządzania projektem.



ROLE W SCRUM

- **Zespół**
 - grupa około 7 interdyscyplinarnych osób, które wykonują prace analityków, projektantów, programistów, testerów ...
 - Posiada przypisaną odpowiedzialność za dostarczenie produktu
- Członkowie projektu – tzw. „świnki”
 - Właściciel produktu (w przypadku braku regularnego zaangażowania zaliczany do „kurczaków”)
 - Mistrz
 - Zespół: architekt, programista, analityk, tester, ekspert ...
- Pozostali interesariusze projektu – tzw. „kurczaki”,
 - nie posiadają formalnej odpowiedzialności
 - nie są częścią procesu Scrum, ale są zainteresowani projektem (klienci, sprzedawcy, decydenci organizacji środowiska wytwarzania)
 - biorą udział w spotkaniach przeglądu sprintu,
 - podczas spotkań nie mają prawa mówić, co zespół powinien zrobić



GŁÓWNE ARTEFAKTY SCRUM

- **BACKLOG (lista zaległości) produktu – rejestr produktowy**
- Lista wymagań funkcjonalnych i нефункциональных projektu, ułożonych według priorytetów z przewidywanym czasem na ich zakończenie i wdrożenie.
- Szacunki są podawane w dniach i są bardziej precyzyjne im wyższa pozycja w kolejce zaległości produktu.
- Priorytety pozycji powinny być ustalone na podstawie największej wartości dla firmy, obliczone za pomocą metody ROI.
- Lista w trakcie realizacji rozwija się i zmienia.
- **BACKLOG sprintu – rejestr zaległości sprintu**
- rejestr zadań, które zespół Scrum zobowiązuje się wykonać całkowicie w bieżącym sprincie.
- Pozycje zaległości sprintu pochodzą z rejestru zaległości produktu.
- Zespół kieruje się priorytetami ustalonymi przez właściciela produktu i swoimi oszacowaniami dotyczącymi czasu ich wykonania.
- Krytyczne jest to, że to sam zespół wybiera pozycje z listy zaległości, ponieważ to zespół zobowiązuje się do zakończenia tych zadań
- Burn down – wykres prezentujący pozostałą pracę do zakończenia sprintu



ITERACJA W SCRUM - SPRINT

- **Sprint**
- Sprint jest to czas przeznaczony na opracowanie jednego przyrostu produktu. Jest to metoda „time boxing: czas, koszt i zakres. Ważniejszy jest zakres niż poślizg w dacie zakończenia.
- Sprint obejmuje projektowanie, programowanie, testowanie i dokumentowanie.
- Tylko po rozpoczęciu sprintu zespół może dodawać lub usuwać elementy z rejestru zaległości sprintu.
- Jeżeli osiągnięcie celu sprintu nie ma sensu, to następuje tzw. nienormalne zakończenie sprintu.

- **Sprint wydania**
- Wydanie oprogramowania do działania (produkcji) wymaga specjalnego sprintu, nazywanego „sprintem wydania”.
- Do sprintu jest dedykowany zespół sprintu wydania

- **Spotkanie planowania sprintu**
- Celem jest ustalenie rejestru zaległości sprintu
- Właściciel produktu opisuje właściwości o najwyższym priorytecie, a zespół decyduje, do czego może się zobowiązać i co może dostarczyć w danym sprincie. Odbywa się to w ciągu dwóch kolejnych spotkań (4 godziny),
- Zespół planuje zadania do wykonania, których zbiór tworzy rejestr zaległości sprintu



CODZIENNY SCRUM

- Codzienne spotkanie trwające 15 minut, ważne jest aby odbywało się codziennie w tym samym miejscu i czasie.
- Podczas spotkania zespół Scrum siedzi w kręgu naprzeciwko siebie i każdego członka zespołu dotyczą następujące trzy pytania:
 1. Co zrobiłeś od ostatniego Scrum?
 2. Co będziesz robił od teraz do następnego Scrum?
 3. Jakie problemy przeszkadzają Ci w wykonywaniu pracy?



PRZEGLĄD SPRINTU

- Na koniec każdego sprintu odbywa się spotkanie przeglądowne sprintu. Podczas tego spotkania zespół Scrum demonstruje, co zostało zakończone w fazie tego sprintu. Zwykle jest to forma prezentacji nowych funkcji.
- Zaleca się, aby spotkania przeglądowne sprintu było nieformalne, z zasady zakazuje się slajdów programu PowerPoint i pozwala na poświęcenie nie więcej niż dwie godziny czasu na przygotowanie się do tego spotkania. Spotkania nie mogą stać się uciążliwe dla zespołu, powinny być naturalnym wynikiem sprintu.
- Uczestnicy spotkania przeglądownego sprint: właściciel produktu, zespół Scrum, Mistrz, zarządzanie, klienci i inżynierowie z innych projektów.
- Podczas tego spotkania projekt jest porównywany z celem sprintu ustalonym podczas spotkania planowanie sprintu. Najlepiej jest, gdy zespół zakończy każdą pozycję z rejestru produktów dla tego sprintu, ale ważniejsze jest osiągnięcie celu sprintu.



RETROSPEKTYWA SPRINTU

- To spotkanie jest wykorzystywane przez Mistrz do omówienia zakończonego Sprintu i ma na celu określenie, co można zmienić w następnym Sprincie, by praca była przyjemniejsza i bardziej produktywna.
- Dyskusja może dotyczyć wszystkiego, co wpływa na pracę zespołu: procesy, praktyki, komunikację, środowisko, narzędzia.
- Scrum powinien być postrzegany jako ramy, które powinny być odpowiednio dostosowane do danego projektu, zespołu i konkretnej sytuacji.
- Retrospektywa Sprint jest ważnym narzędziem, które pozwala zespołowi na ciągłą poprawę w całym cyklu życia projektu.



KORZYŚCI Z ZASTOSOWANIA METODY SCRUM

- Zapewnia najwyższe wartości firmy na wczesnym etapie projektu, unikając jednocześnie nierealistycznych wymagań i niepotrzebnej pracy (eliminacja marnotrawstwa)
- Poprawia satysfakcję klienta
- Dostarcza podejścia kierowanego przez klienta
- Skupia się na szybkości dostawy
- Zapewnia otwartość i przejrzystość dla klientów
- Usuwa przeszkody w sposób priorytetowy i systematyczny
- Poprawia utrzymanie pracowników przez umożliwienie oraz promowanie samorządności, komunikacji w zespole, naukę i rozwój

- Efekty firmy Microsoft ze Scrum: czterokrotny wzrost średniej wydajności i dwanaście razy lepsza jakość



PROGRAMOWANIE EKSTREMALNE

- XP próbuje zredukować koszty zmian wymagań poprzez wiele krótkich cykli wytwarzania oprogramowania, zamiast jednego jak np. w kaskadowym modelu.
- Według tej metodyki zmiany są naturalnym, nieuniknionym i wskazanym aspektem projektu oraz powinny być włączone w plany projektu, zamiast próby stworzenia niezmiennego zestawu wymagań.



PROGRAMOWANIE EKSTREMALNE

Cztery podstawowe czynności:

1. Tworzenie kodu źródłowego

2. Testowanie

- Testy jednostkowe
- Testy akceptacyjne

3. Słuchanie

4. Projektowanie



PROGRAMOWANIE EKSTREMALNE

WARTOŚCI

- Komunikacja
- Prostota
- Opinie zwrotne (Feedback)
- Odwaga
- Szacunek



PORÓWNANIE METODOLOGII

| | Kaskadowy | Spiralny | Iteracyjny | Scrum |
|------------------------------------|-------------------------------|-------------------------------|-----------------------------|----------------------------------|
| Zdefiniowany proces | wymagane | wymagane | wymagane | wymagane |
| Produkt finalny | Ustalony w trakcie planowania | Ustalony w trakcie planowania | Ustalony w trakcie projektu | Ustalony w trakcie projektu |
| Koszt projektu | Ustalony w trakcie planowania | częściowo zmienny | Ustalony w trakcie projektu | Ustalony w trakcie projektu |
| Data zakończenia | Ustalony w trakcie planowania | częściowo zmienny | Ustalony w trakcie projektu | Ustalony w trakcie projektu |
| Reagowanie na zmianę potrzeb rynku | Na etapie planowania | Głównie na etapie planowania | Na końcu każdej iteracji | Na każdym etapie |
| Elastyczność zespołu, kreatywność | Ograniczona | Ograniczona | Ograniczona | Nieograniczona podczas iteracji |
| Transfer wiedzy | Trening przed projektem | Trening przed projektem | Trening przed projektem | Praca zespołowa podczas projektu |
| Prawdopodobieństwo sukcesu | Niskie | Średnio niskie | Średnie | Wysokie |

Schwaber, K. (1997). Scrum development process. In *Business object design and implementation* (pp. 125). Springer, London.