Wykorzystanie metod i języków formalnych w budowie systemów GIS z uwzględnieniem języka UML

Tomasz Kubik, Ziemowit Balcer

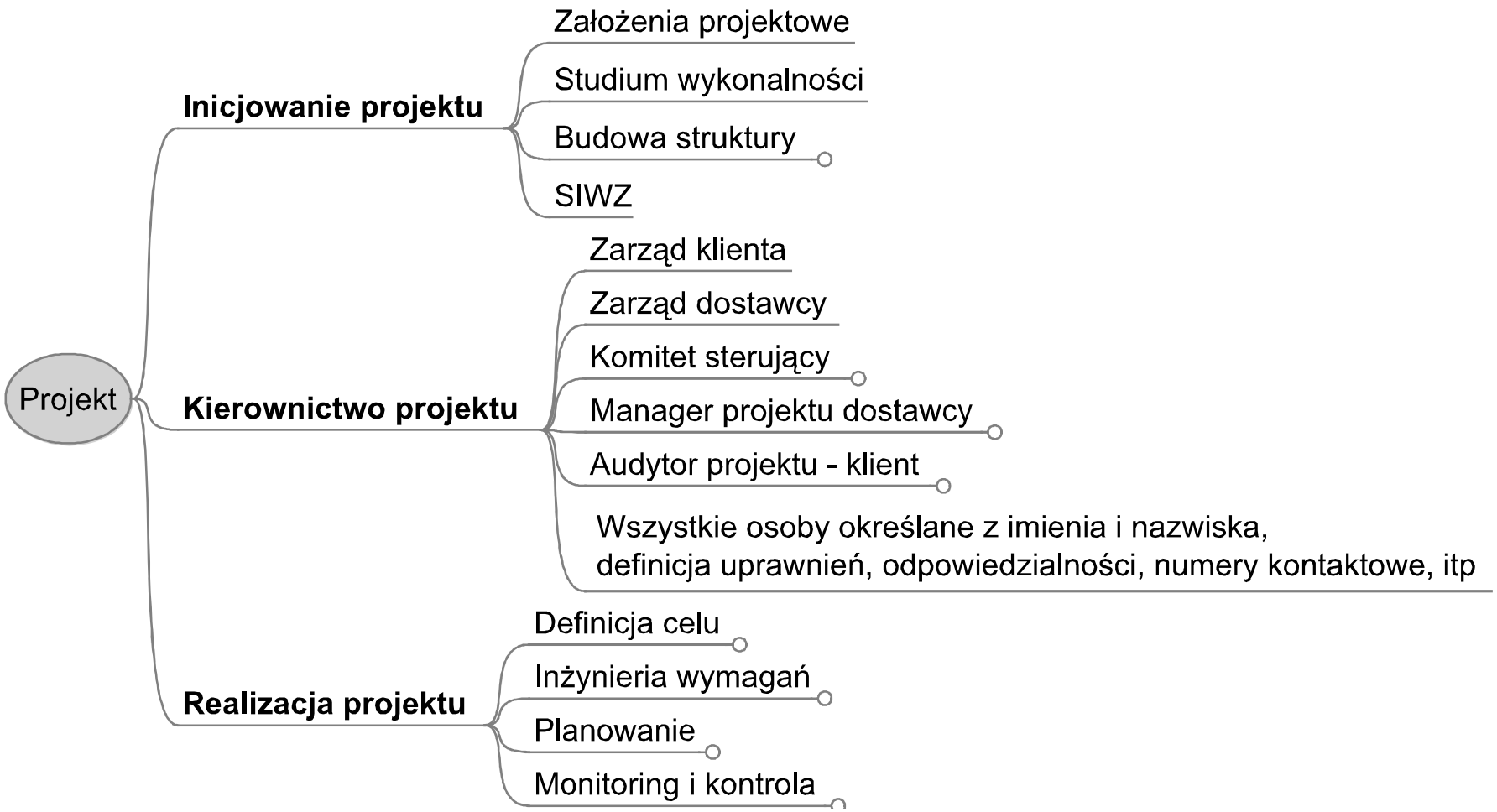
Materiały do seminarium zorganizowanego
w Instytucie Geodezji i Geoinformatyki
Uniwersytetu Przyrodniczego we Wrocławiu

19-20 października 2009

System informatyczny

- Zbiór powiązanych ze sobą elementów, którego funkcją jest przetwarzanie danych przy użyciu techniki komputerowej.
- Na systemy informatyczne składają się obecnie takie elementy jak:
  - sprzęt (komputery, urządzenia służące do przechowywania danych, do komunikacji między sprzętowymi elementami systemu, do komunikacji między ludźmi a komputerami, sensory i elementy wykonawcze, urządzenia służące do przetwarzania danych nie będące komputerami)
  - oprogramowanie
  - ludzie
  - elementy organizacyjne (procedury)
  - elementy informacyjne (bazy wiedzy)

Projekt systemu i jego realizacja



Studium wykonalności

- Decyduje o tym, czy warto wykonać system
- Odpowiada na pytania
  - Czy system przyczyni się do realizacji celów organizacji przy zachowaniu wszystkich uwarunkowań techniczno-ekonomiczno-prawnych?
  - Jakie problemy aktualnie występują i czy system pomoże w ich rozwiązaniu?
  - Które z problemów są krytyczne, a które nie?
  - Czy system może być stworzony przy użyciu dostępnych technologii w ramach określonego budżetu i ograniczeń czasowych?
  - Czy system może być zintegrowany z istniejącymi systemami, a jeśli tak, to jakie będą problemy z integracją?
  - Co się stanie jeśli system nie powstanie?

Etapy budowy systemu informatycznego dla przedsiębiorstwa

1 *To, co klient zamówił*

2 *To, co analityk zrozumiał.*

3 *To, co opisywał projekt.*

4 *To, co wykonali programiści.*

5 *Projekt po uruchomieniu i wdrożeniu.*

6 *To, za co klient zapłacił.*

7 *A to, czego klient potrzebował*

8 *Praktyczne zastosowanie projektu.*

System informacyjny projektu

- Dotyczy następujących obszarów:
  - infrastruktury komunikacyjnej
  - zbierania informacji
    - dokumentacji projektowej, korespondencji, notatek
    - raportów o stanie projektu
    - raportów o wydajności projektu
    - raportów o sytuacjach wyjątkowych
  - dystrybucji informacji
  - autoryzacji i archiwizacji

Informatyczne wsparcie systemu informacyjnego

- Systemy zarządzania dokumentami (DMS)
- Systemy kontroli wersji (CVS i SVN)
- Edytory map myśli (mind maps)
- Systemy zarządzania pracą grupową
- Zarządzanie projektami

Przykłady

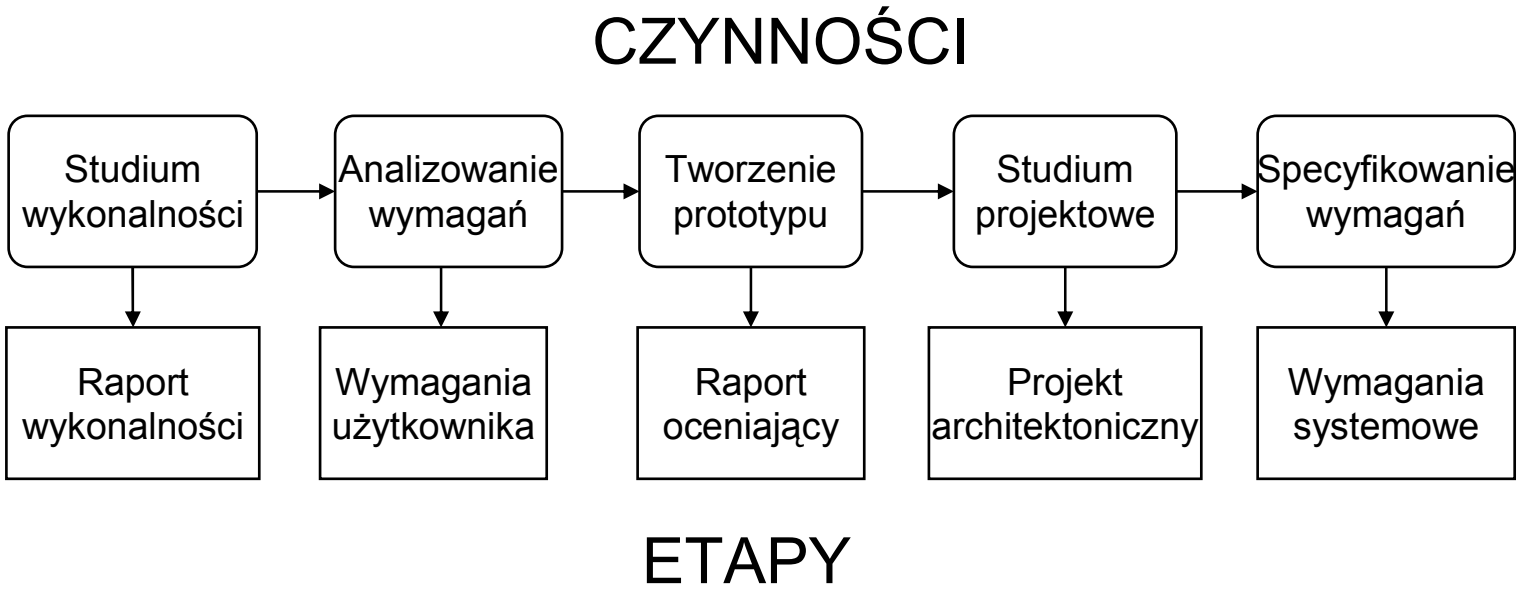
Inżynieria wymagań

- Proces określania oczekiwań użytkownika co do usług systemu i ograniczeń, które system musi spełniać, zarówno podczas tworzenia, jak i podczas działania
- Wymagania
  - Opisują usługi, które system ma udostępniać i jego ograniczenia
  - Mówią, co system powinien robić, zaś projekt – jak
  - Są definiowane na różnych poziomach abstrakcji: od ogólnego opisu usług systemu po precyzyjne matematyczne definicje realizowanych przez niego funkcji
- Specyfikacja wymagań dla danego systemu
  - zawiera zarys jego architektury, pełniąc rolę dokumentu referencyjnego dla pozostałych specyfikacji projektu
  - określa współdziałanie systemu z innymi systemami, co wymusza uwzględnienie dodatkowych ograniczeń bądź rozszerzeń w projekcie
  - forma realizacji projektu może być zewnętrznym wymaganiem systemowym

Definicje wymagań

- Mogą mieć różną formę, w zależności od celu jakiemu służą:
  - Formę otwartą, gdy występują podczas ustalania kontraktu
  - Formę zamkniętą, gdy stanowią podstawę do implementacji
- Pożądane cechy definicji:
  - Spójność (bez sprzeczności w zdefiniowanych oczekiwaniach)
  - Kompletność (obejmujące wszystkie wymagane oczekiwania)
  - Ścisłość (zawężające możliwy zakres ich interpretacji)
  - Realność (tzn. realizowalne)
  - Weryfikowalność (dające się zweryfikować)
  - Istotność (najlepiej spełniające potrzeby użytkownika)

Kamienie milowe w procesie definiowania wymagań



Typy wymagań

- **Wymagania użytkownika**
  - Definiują oczekiwania co do usług oferowanych przez system oraz ograniczeń, w jakich system ma działać w sposób ogólny
  - Mieszczą w swoim zakresie wymagania funkcjonalne i niefunkcjonalne
  - Ich zapis odbywa się w języku naturalnym, z wykorzystaniem formularzy i intuicyjnych diagramów, w sposób zrozumiały dla użytkowników systemu nie posiadających wiedzy technicznej
  - Powstają na podstawie wywiadu przeprowadzonego z klientem
- **Wymagania systemowe**
  - Precyzyjnie opisują usługi i ograniczeń systemu
  - Mogą być przedstawione za pomocą modeli
  - Pojawiają się jako część kontraktu pomiędzy klientem a dostawcą
- **Specyfikacja projektu oprogramowania**
  - Jest to abstrakcyjny opis projektu oprogramowania, dający podstawę do utworzenia szczegółowego opisu projektu i implementacji
  - Opis ten powstaje na potrzeby programistów

Podział wymagań

- **Wymagania funkcjonalne**
  - Jakie usługi ma oferować system
  - Jak ma reagować na określone dane wejściowe
  - Jak ma się zachowywać w określonych sytuacjach
  - Czego nie powinien robić
- **Wymagania niefunkcjonalne**
  - Ograniczenia usług i funkcji systemu, np. ograniczenia czasowe, ograniczenia dotyczące procesu tworzenia, standardy itd.
- **Wymagania dziedzinowe**
  - Pochodzą z dziedziny zastosowania
  - Mogą być funkcjonalne lub niefunkcjonalne.

Wymagania funkcjonale

- Definiują oczekiwania co do funkcjonalności lub usług systemu
- Zależą od rodzaju tworzonego oprogramowania, spodziewanych użytkowników oprogramowania i rodzaju wytwarzanego systemu.
- Ich poziom abstrakcji i szczegółowość może być różny
  - wymagania użytkownika definiowane są w sposób ogólny
  - wymagania funkcjonalne systemowe definiują funkcje systemu, jego wejścia, wyjścia, wyjątki itd. w sposób szczegółowy

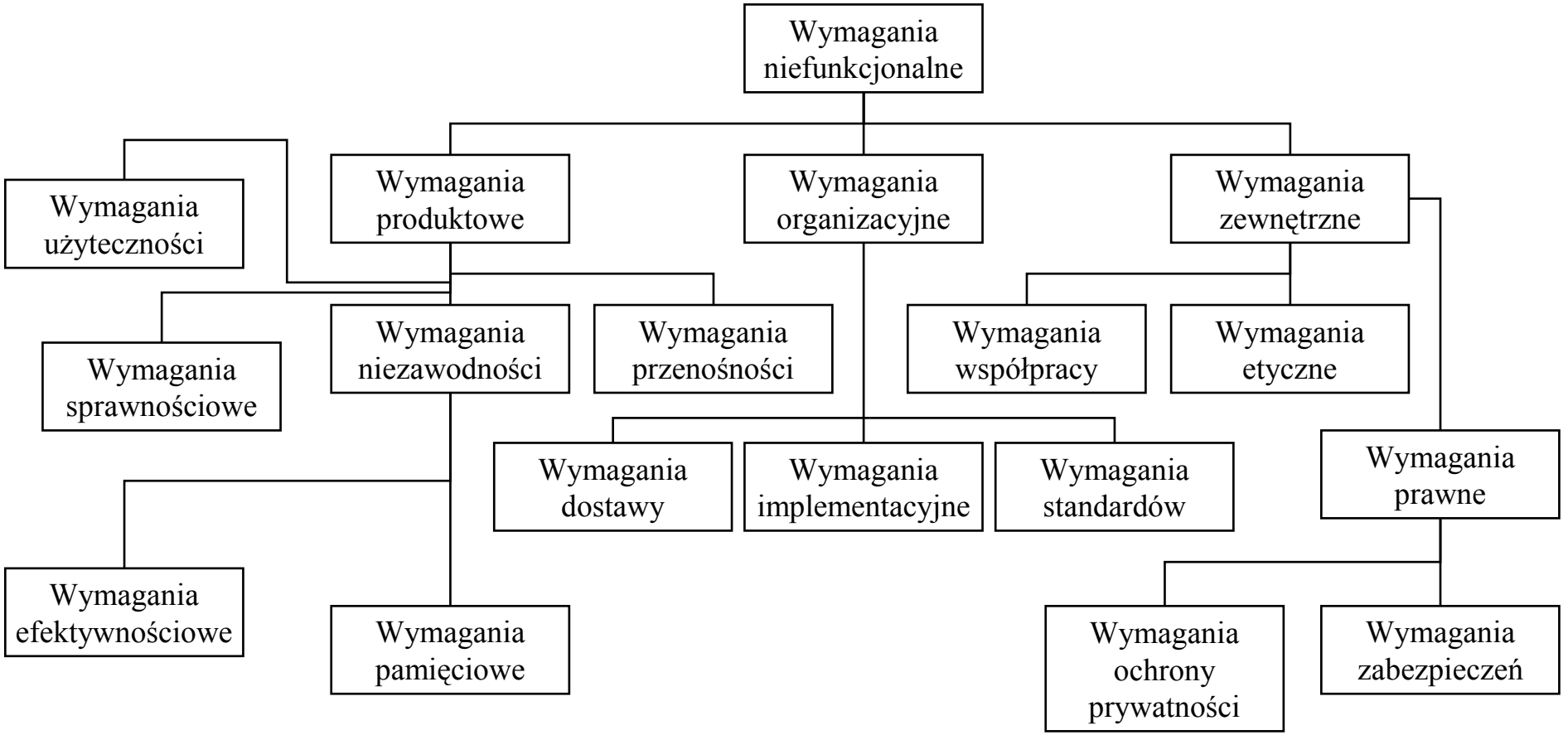
Wymagania niefunkcjonalne

- Opisują cechy i ograniczenia systemu nie związane z jego funkcjonalnością
  - Wielkość zasobów, czas odpowiedzi, skalowalność, bezpieczeństwo, niezawodność, …
- Mogą dotyczyć oprogramowania narzędziowego
  - System operacyjny, język programowania, metodologia projektowa, …
- Mogą być istotniejsze od funkcjonalnych, bo ich niespełnienie może uniemożliwić pracę systemu
- Ich doprecyzowanie czasem jest trudne, dlatego stosuje się definicje celów oraz przyjmuje pewne miary

Podział wymagań niefunkcjonalnych

- Wymagania **produktowe**
  - Określają zachowanie produktu (szybkości działania systemu, wymagane zasoby, niezawodność, …)
- Wymagania **organizacyjne**
  - Wynikają ze strategii i procedur zaimplementowanych u klienta i dostawcy (standardy procesu, wymagana dokumentacja, metodologia projektowania, …)
- Wymagania **zewnętrzne**
  - Wynikające z czynników zewnętrznych dla systemu i procesu jego tworzenia (powiązanie z innymi systemami, uwarunkowania prawne, …)

Typy wymagań niefunkcjonalnych



Kryteria jakości

- Zestaw cech, które opisują jakość produktu lub na podstawie których jego jakość jest oceniana
- Norma ISO 9126
  - Funkcjonalność
  - Niezawodność
  - Użyteczność
  - Wydajność
  - Pielęgnowalność
  - Przenośność

model McCalla

Działanie programu

Przyjazność	Efektywność użytkowania programu i przyjazność jego interfejsu
Bezpieczeństwo	Bezpieczeństwo użytkowania programu pod kątem kontroli uprawnień do korzystania z niego oraz odporności na skutki nieprawidłowej obsługi
Wydajność	Ocena wydajności systemu i sposobów zarządzania zasobami
Poprawność	Stopień realizacji wymagań, kompletność i logiczność wdrożenia, zgodność działania programu ze specyfikacją
Niezawodność	Stopień odporności programu na błędy, jego poprawność formalna oraz sposoby reakcji na błędne sytuacje

model McCalla

Przystosowanie do modyfikacji

Pielęgnowalność	Stopień przystosowania programu do poprawy, modyfikacji, rozszerzania, adaptowania
Elastyczność	Możliwości rozbudowania programu o nowe funkcje oraz uniwersalność wdrożonych rozwiązań
Testowalność	Przystosowanie do procesu testowania oraz instrumentacja tego procesu

Mobilność oprogramowania

Przenośność	Zdolność do łatwego uruchomienia na innych maszynach lub systemach oprogramowania
Uniwersalność	Możliwość wykorzystania istniejącego oprogramowania lub jego fragmentów do konstrukcji innych programów
Testowalność	Przystosowanie do procesu testowania oraz instrumentacja tego procesu

Wymagania dziedzinowe

- Wynikają z dziedziny zastosowanie systemu
- Mogą być funkcjonalne lub niefunkcjonalne
- Formułowane są głównie w języku ekspertów,
  - zdarza się, że brakuje w nich kluczowych informacji uznanych przez ekspertów za oczywiste
- Często określają sposób przetwarzania danych za pomocą wzorów matematycznych, np.:
  - transformacja danych pomiędzy układami odniesień przestrzennych powinna odbywać się wg wzoru:

Wymagania systemowe

- Bardziej szczegółowe opisy wymagań użytkownika
- Mogą być podstawą kontraktu na implementację systemu
  - Powinny być pełną i niesprzeczną specyfikacją całego systemu
- Określają punkt wyjścia do projektowania systemu
- Specyfikacja wymagań systemowych może zawierać różne modele systemu

Specyfikacja interfejsów

- Większość systemów musi współdziałać z innymi systemami, które już znajdują się w środowisku
- Musi istnieć specyfikacja interfejsów istniejących systemów
- Specyfikacja interfejsów zawierać może dotyczyć trzech typów informacji
  - Interfejsów proceduralnych
  - Struktur danych przekazywanych między podsystemami
  - Reprezentacje danych (na poziomie binarnym)

Definiowania wymagań

- Z użyciem języka naturalnego
  - Jest elastyczne (pozwala wyrażać to samo na wiele sposobów)
  - Może być niejednoznaczne (przy różnej interpretacja tych samych słów).
  - Trudno poddaje się systematyzowaniu (podziale na kategorie)
- Powinno odbywać się wg ustalonej procedury
  - Przy użyciu jednoznacznych pojęć oraz zwrotów „będzie” i „powinien”
  - Bez użycia żargonu komputerowego
  - Z zastosowaniem wyróżnień, rozwinięć, wyliczeń itp.
- Powinno kończyć się opracowaniem specyfikacji
  - Z opisem definiowanego fragmentu systemu, realizowanej funkcji itp.
  - Z opisem danych wejściowych i wyjściowych, ze wskazaniem ich źródła i miejsca ich wykorzystania
  - Z opisem powiązań z innymi częściami systemu niezbędnych do realizacji danego wymagania
  - Z opisem warunków początkowych i końcowych, które muszą być spełnione przy realizacji wymagań funkcjonalnych
  - Z opisem efektów ubocznych operacji (jeśli występują)

Języki definiowania wymagań

Notacja	**Opis**
Strukturalny język naturalny	Wymagania są redagowane z użyciem standardowych formularzy i szablonów
Języki opisu projektu	Wymagania są wyrażany w języku abstrakcyjnym, zbliżonym do języka programowania, zgodnie z przyjętym w projekcie modelem
Notacja graficzna	Wymagania wyobrażane są za pomocą symboli graficznych z tekstowymi dopiskami.
Formalizmy matematyczne	Wymagania definiowane są w sposób formalny, w języku matematyki (np. algebry zbiorów czy symboli opisujących maszyny stanów skończonych).

Formularze

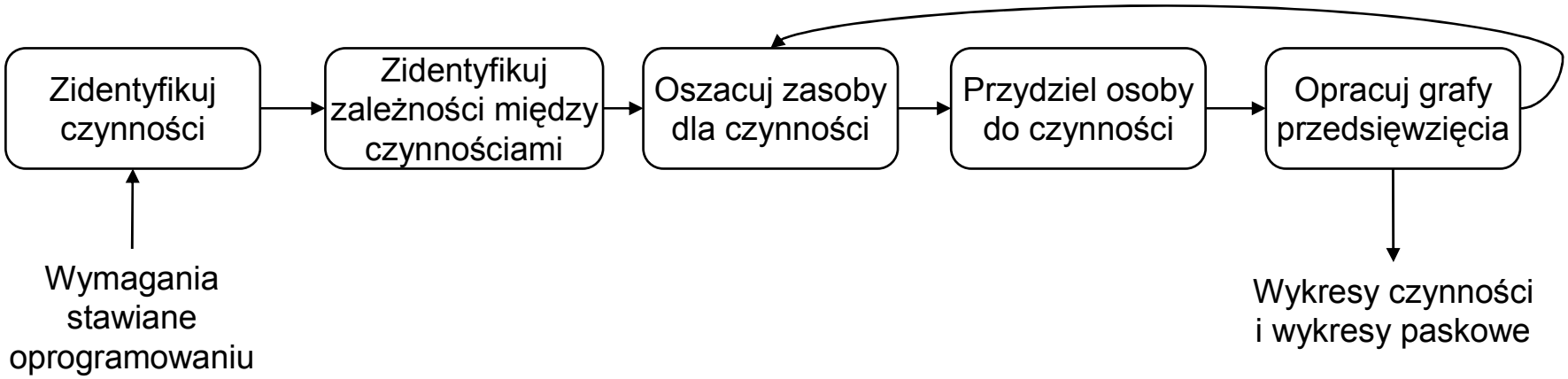
- Mogą mieć format tabeli bądź wyliczenia zawierającego
  - Definicję funkcji lub wejścia
  - Opis danych wejściowych i źródła pochodzenia
  - Opis danych wyjściowych i ich celu
  - Wymagania dodatkowe i powiązania
  - Warunki wstępne i wyjściowe
  - Efekty uboczne

Język definiowania interfejsów

- Język opisu programów, zawierający instrukcje zwiększające wyrazistość opisu
- Traktowany raczej jako część projektowa, a nie specyfikacja wymagań
- Jest on stosowany, gdy:
  - Opisywana funkcja jest ciągiem akcji o istotnej kolejności wykonania
  - Istnieje konieczność określenia interfejsów programowych lub sprzętowych
- Nie pozwala określić wymagań dziedzinowych

```
interface SerwerDrukowania {
// definiuje abstrakcyjny serwer drukowania
// wymaga:    interface Drukarka, interface DokumentDoWydruku
// udostepnia: inicjuj, drukuj, wyświetlKolejkeZadań, anulujZadanieDrukowania,
// zmieńDrukarkę
                    void inicjuj ( Drukarka d ) ;
                    void drukuj ( Drukarka d, DokumentDoWydruku w) ;
                    void wyświetlKolejkęZadań ( Drukarka d ) ;
                    void anulujZadanieDrukowania (Drukarka d, DokumentDoWydruku w) ;
                    void zmieńDrukarkę(Drukarka d1, Drukarka d2, DokumentDoWydruku w) ;
} //SerwerDrukowania
```

Planowanie

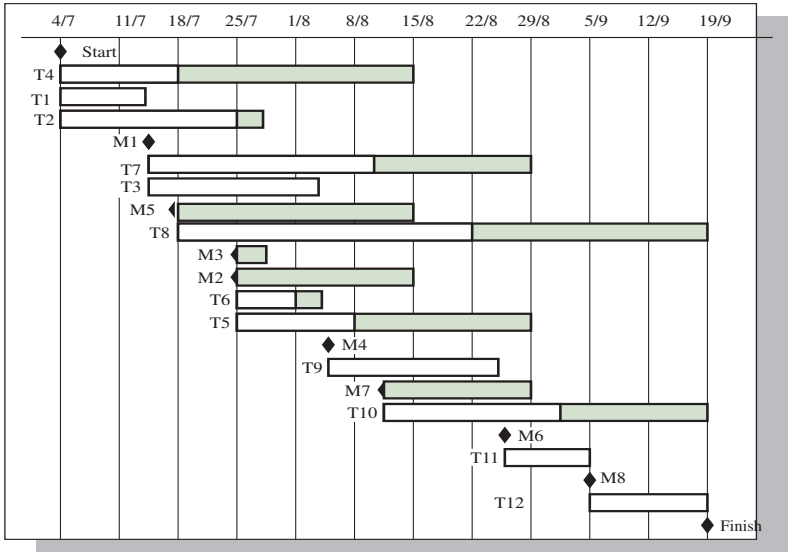


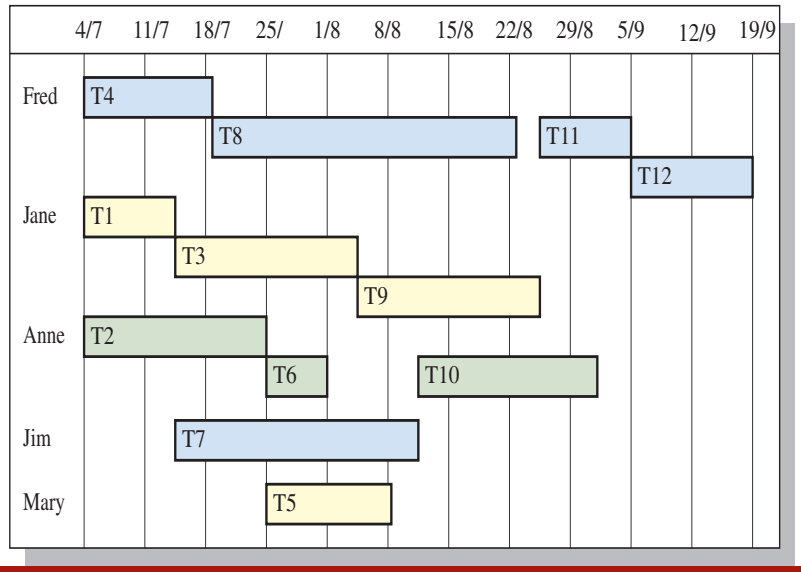
- Problemy
  - Istnieją trudności w szacowaniu złożoności problemów i czasu koniecznego na ich rozwiązanie
  - Produktywność nie jest proporcjonalna do liczby pracowników
  - Dodawanie pracowników spowalnia przedsięwzięcie ze względu na czas komunikacji
  - Mogą pojawić się zdarzenia niespodziewane, na które należy przewidzieć margines czasowy

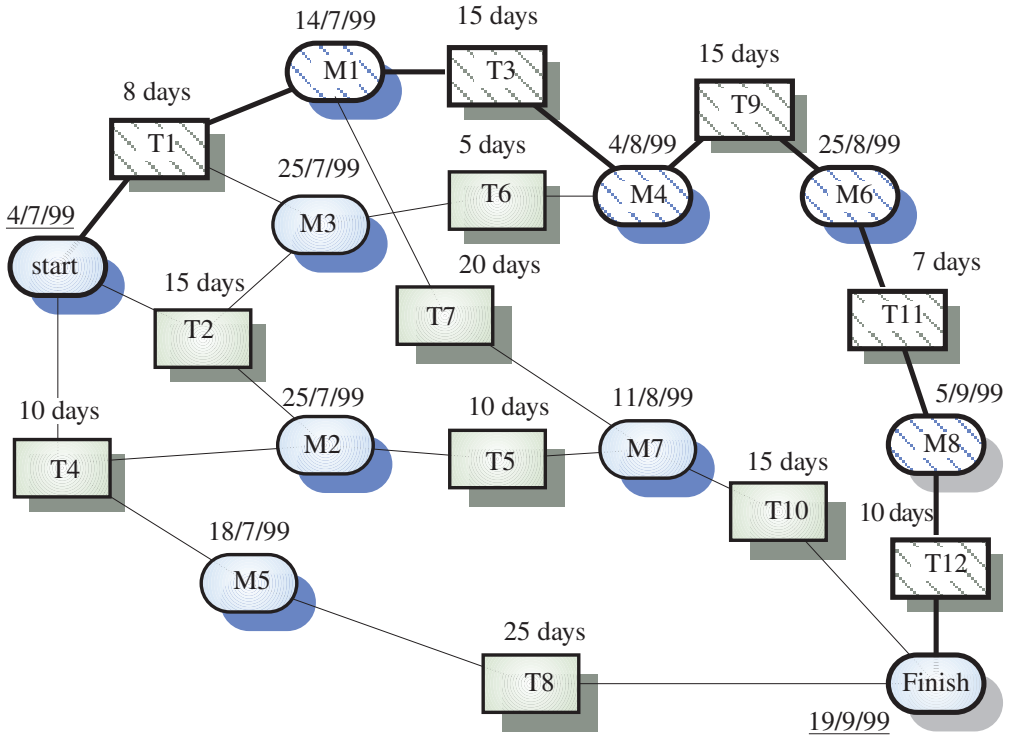
Harmonogramy

- Odnoszą się do konkretnych uwarunkowań związanych z czasem, kosztem i środkami koniecznymi do realizacji zadań
  - daty rozpoczęcia i zakończenia
  - warunków rozpoczęcia i zakończenia
  - wykorzystywania zasobów, w tym osób odpowiedzialnych
  - kosztu i pracochłonności
- Wizualizowane przez
  - sieć działań, pokazującą zależności i ścieżkę krytyczną
  - wykres paskowy, pokazujący harmonogram w terminach dni roku kalendarzowego

Wykres paskowy i sieć działań







Struktura dokumentów

- Przedmowa
- Wstęp
- Słownik
- Definicja wymagań użytkownika
- Architektura systemu
- Specyfikacja wymagań systemowych
- Modele systemu
- Ewolucja systemu
- Dodatki
- Skorowidz

Szablon dokumentów ze specyfikacją wymagań (IEEE)

1. Wstęp
    1.1. Przeznaczenie tej dokumentacji wymagań
    1.2. Zakres produktu
    1.3 Definicje, akronimy i skróty
    1.4. Odnośniki
    1.5. Przegląd pozostałej części dokumentu
2. Ogólny opis
    2.1 Wizja produktu
    2.2 Funkcje produktu
    2.3 Charakterystyka użytkowników
    2.4 Ogólne ograniczenia
    2.5 Założenia i zależności
3. Szczegółowe wymagania
4. Dodatki
5. Skorowidz

Przykładowe dokumenty projektowe

Widoki

- Uczestnicy miewają różne punkty widzenia na ten sam problem
- Analiza z wielu perspektyw jest ważna, ponieważ nie istnieje jedynie słuszna metoda na analizowanie wymagań
- ISO/IEC 10746-1:1998, Basic Reference Model of Open Distributed Processing (ODP) definiuje następujące widoki:
  - korporacyjny, związany z przeznaczeniem, zakresem i regułami działania systemu w organizacji, której jest częścią
  - obliczeniowy, związany z opisem interfejsów i schematów interakcji z usługami
  - informacyjny, związany z semantyką przetwarzania informacji, zorientowany na uzyskanie semantycznej interoperacyjności
  - inżynierski, związany z infrastrukturą wymaganą do rozproszenia systemu, z występującymi w niej mechanizmami oraz realizowanymi funkcjami
  - technologiczny, związany z konkretnymi technologiami wspierającymi implementację systemów rozproszonych, łącznie ze sprzętem i oprogramowaniem

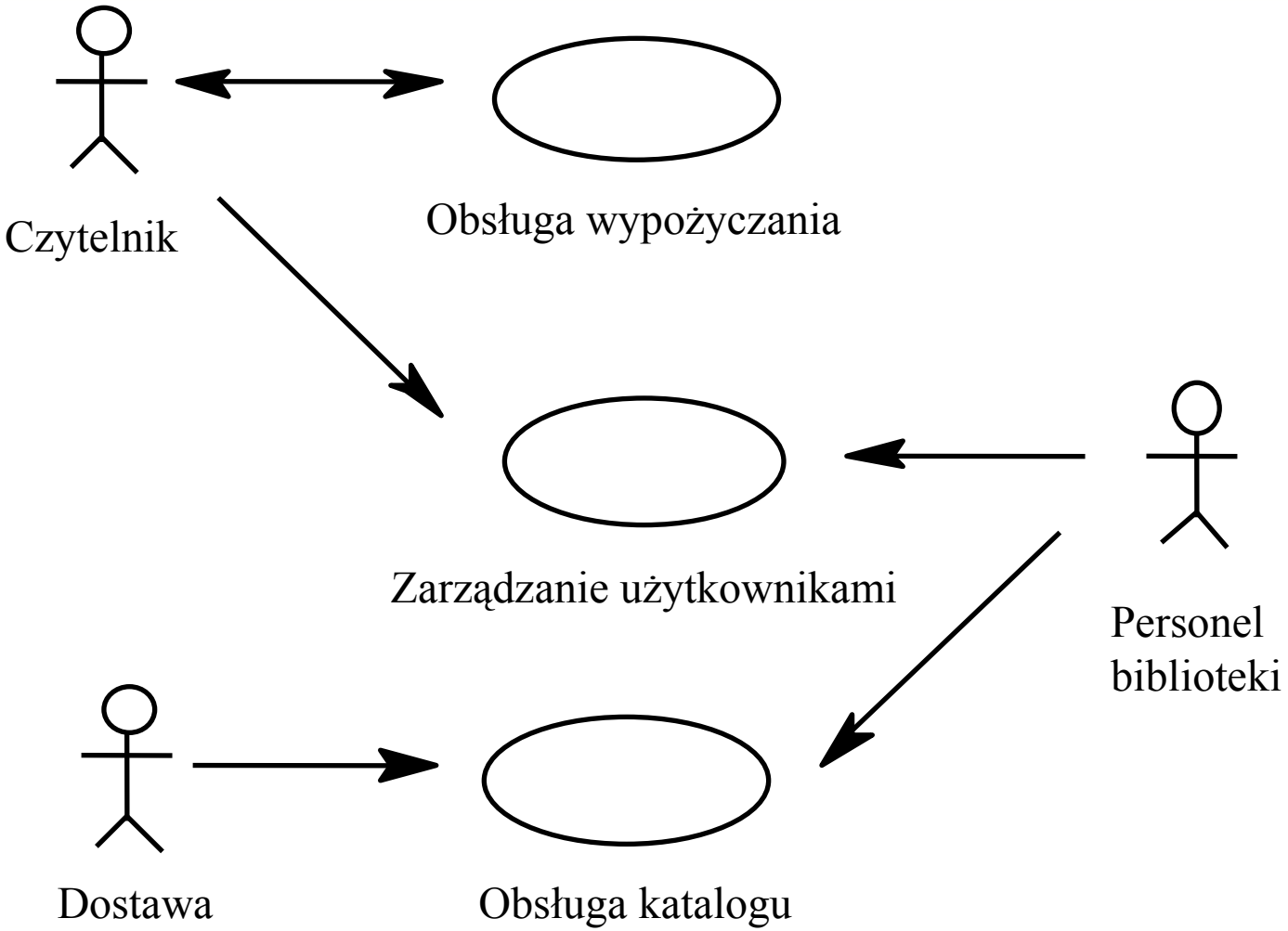
Scenariusze

- Scenariusze są przykładami, jak system jest używany w praktyce
- Są przydatne podczas wydobywania wymagań, ponieważ ludzie rozumieją je dużo lepiej niż abstrakcyjne opisy tego, co oczekują od systemu
- Scenariusze są szczególnie przydatne w dodawaniu szczegółów do ogólnych opisów
- Scenariusze mogą dotyczyć również reakcji systemu na zdarzenia
- Scenariusze obejmują:
  - Stan systemu przed rozpoczęciem scenariusza
  - Normalny przebieg zdarzeń scenariusza
  - Co może pójść źle i jak to jest obsługiwane
  - Inne zdarzenia, które mogą dziać się w tym samym czasie
  - Stan systemu po zakończeniu scenariusza

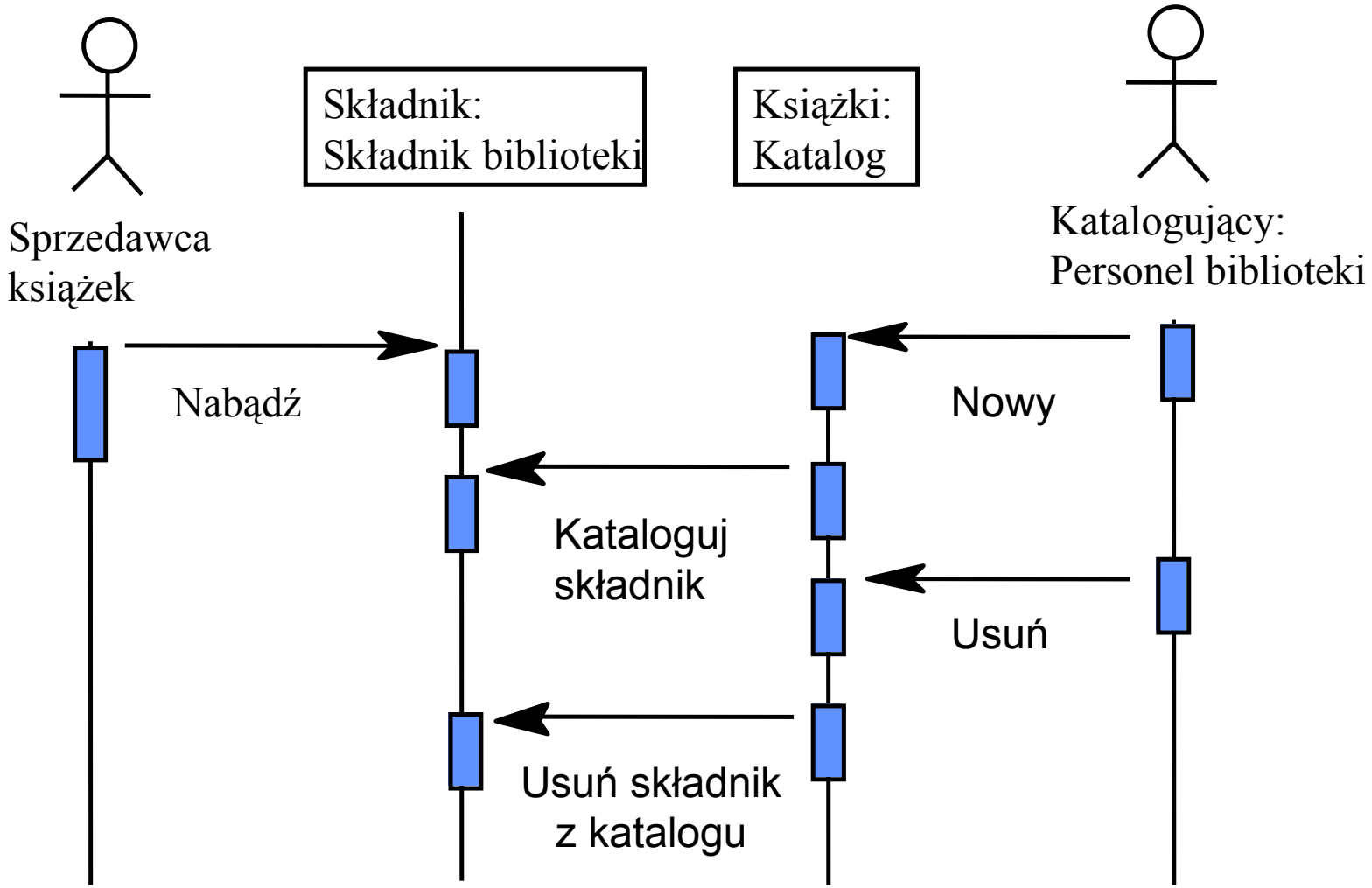
Przypadki użycia

- Przypadki użycia są używaną w UML-u techniką opartą na scenariuszach i definiują aktorów biorących udział w interakcji, co opisuje samą interakcję
- Zbiór przypadków użycia powinien opisywać wszystkie interakcje w systemie
- Diagramy przebiegów mogą służyć do dodawania szczegółowej informacji do przypadków użycia

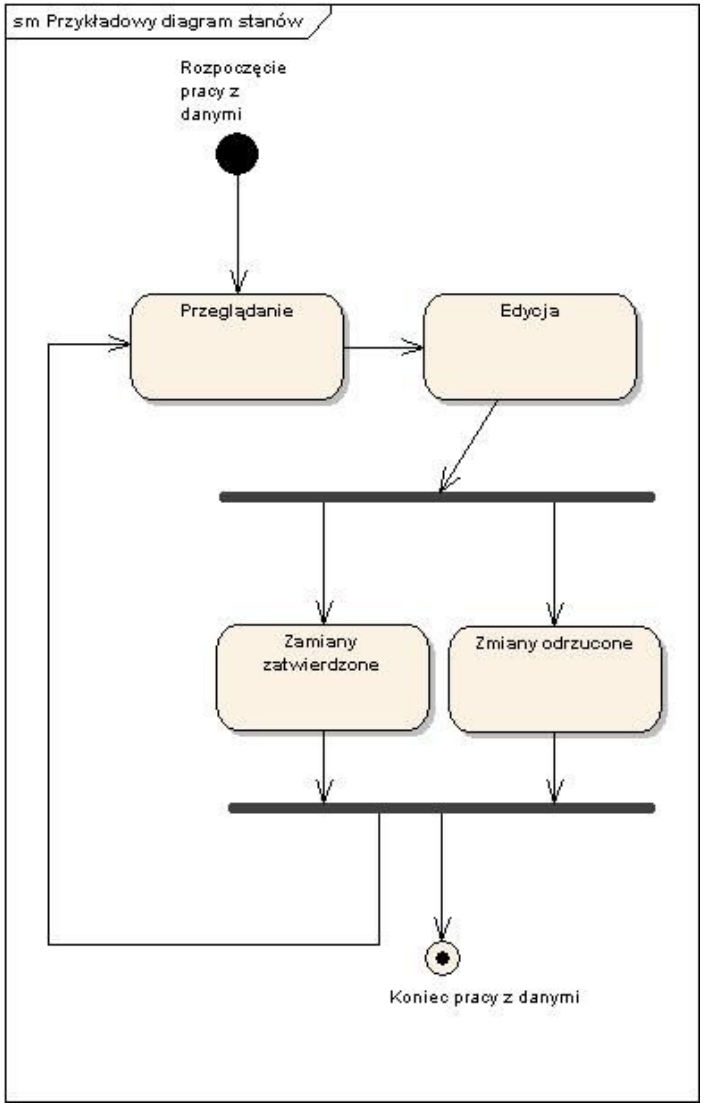
Przykład diagramu przypadków użycia

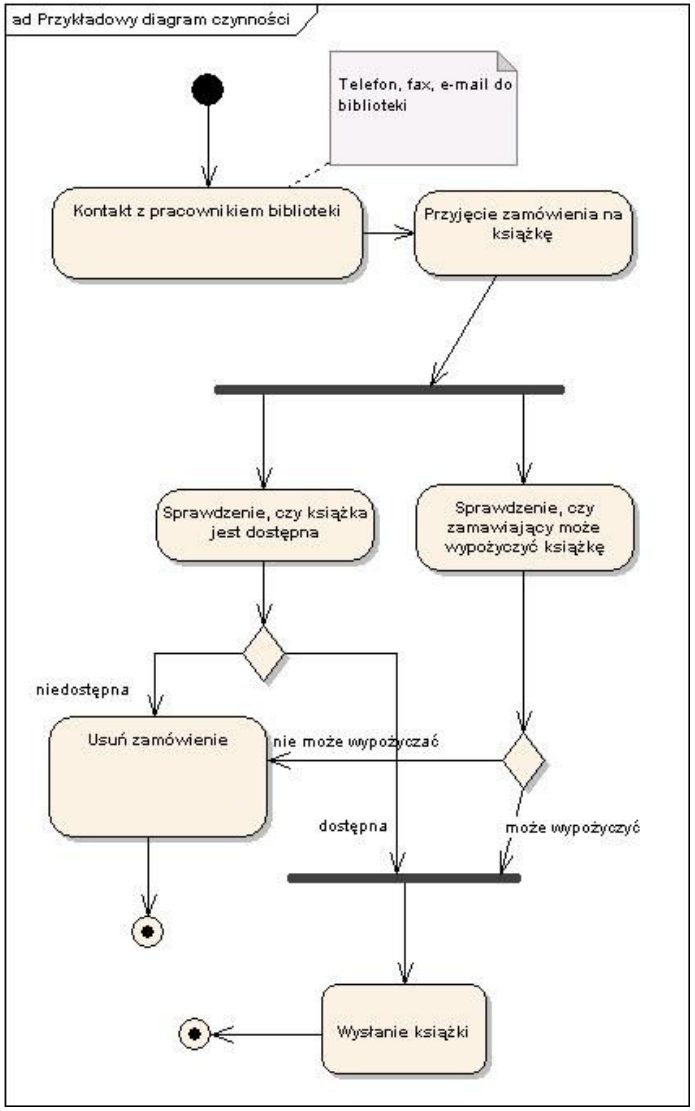


Przykład diagramu przebiegów



Przykład diagramów stanów i czynności





Techniki zatwierdzania wymagań

- Przeglądy wymagań
  - Systematyczne analizy wymagań aż do zatwierdzenia wykonywane przez zespół składający się z pracowników obu stron
  - Mogą być formalne (udokumentowane) lub nieformalne
  - Powinny sprawdzać zrozumiałość wymagań, ich pochodzenie, elastyczność (możliwość zmian i ich wpływ na inne wymagania)
- Prototypowanie
  - Przedstawianie klientom działającego modelu systemu
- Generowanie testów
  - Tworzenie testów dla sprawdzenia czy wymagania są weryfikowalne
- Automatyczne sprawdzanie niesprzeczności
  - Sprawdzanie niesprzeczności wymagań wyrażonych w strukturalnej lub formalnej notacji

Testowanie - definicje

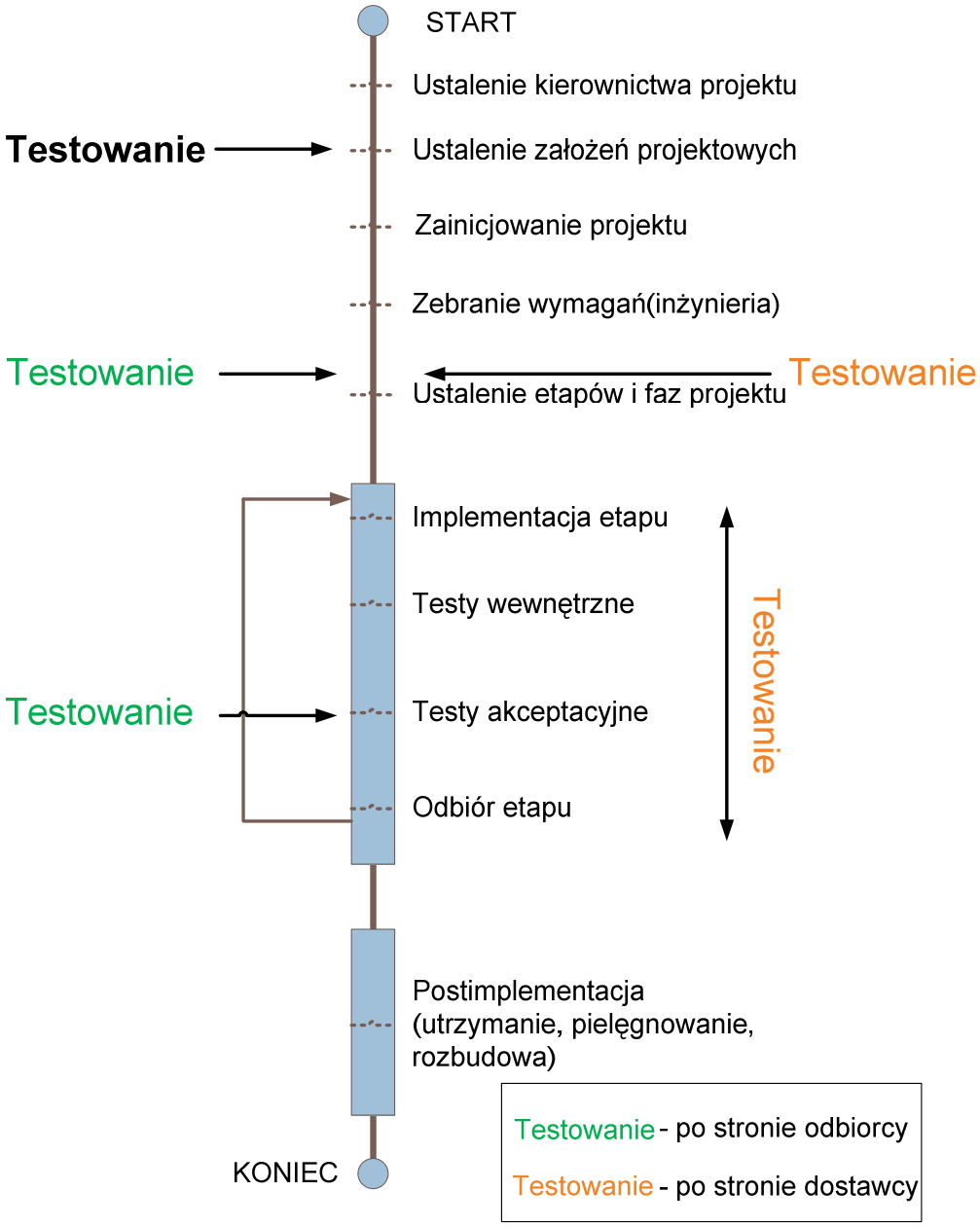
- **Testowanie –** proces sprawdzania/oceny czy produkt lub jego część jest „dobra”
- **Walidacja –** Sprawdzenie oprogramowania pod względem jego użyteczności i sposobu spełnienia potrzeb.
- **Weryfikacja –** Sprawdzenie oprogramowania, czy spełnia wymagania postawione w fazie poprzedniej (zapisane w dokumentach projektowych).

Testowanie – etapy - odbiorca

- Przy ustalaniu założeń projektowych
- Przy szczegółowym ustalaniu wymagań
- Przy odbiorze etapu prac

Linia życia

- Testowanie

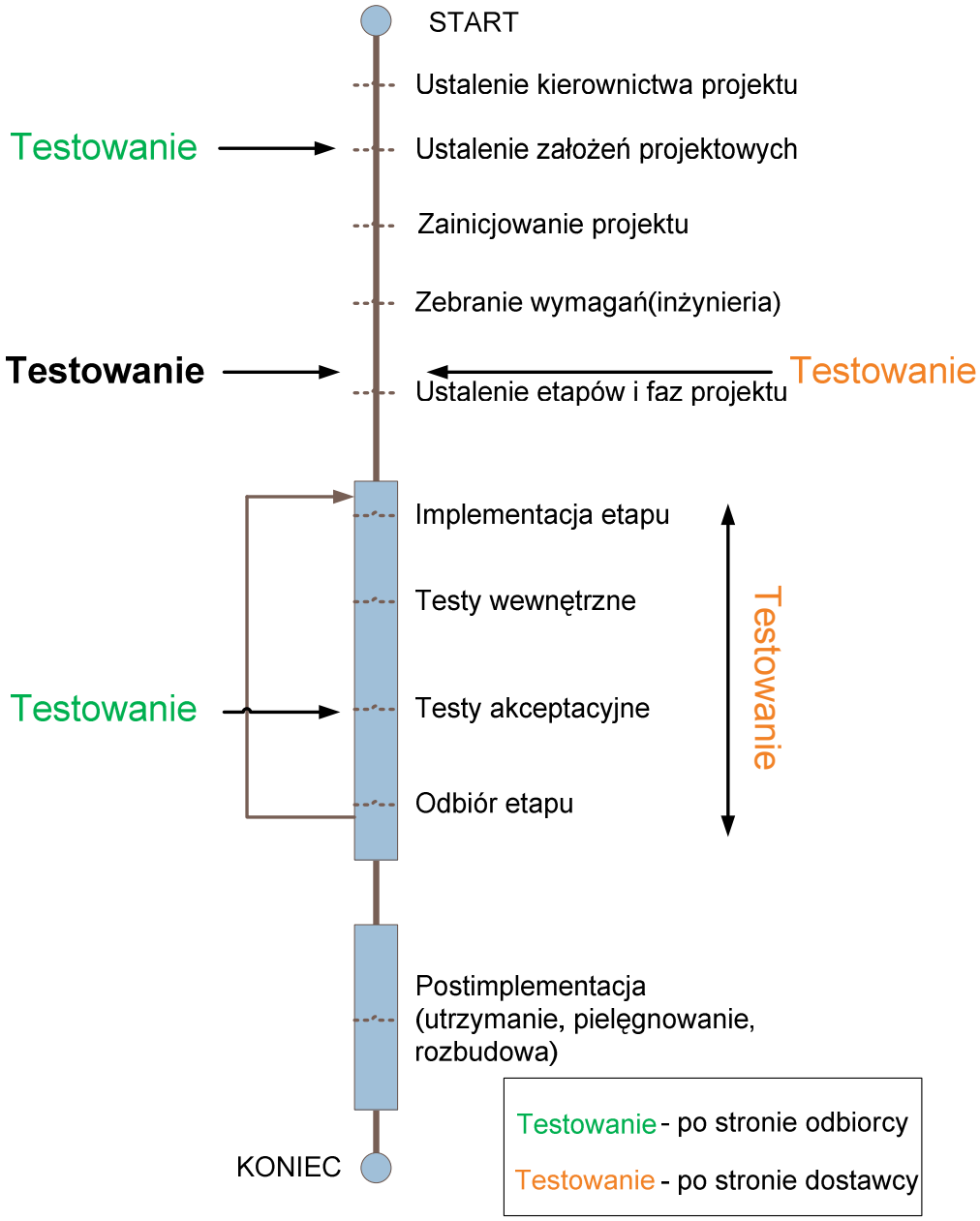


Testowanie – etap założeń projektu

- Określenie sposobu definicji odbioru i weryfikacji wdrażanego przez dostawcę oprogramowania:
  - Żądanie **definicji** i przeprowadzenia **testów akceptacyjnych** według scenariuszy testowych.
  - Żądanie wykorzystywanie **internetowego systemu** wspomagającego komunikację, definicję zagadnień etapu, zgłaszania uwag.

Linia życia

- Testowanie



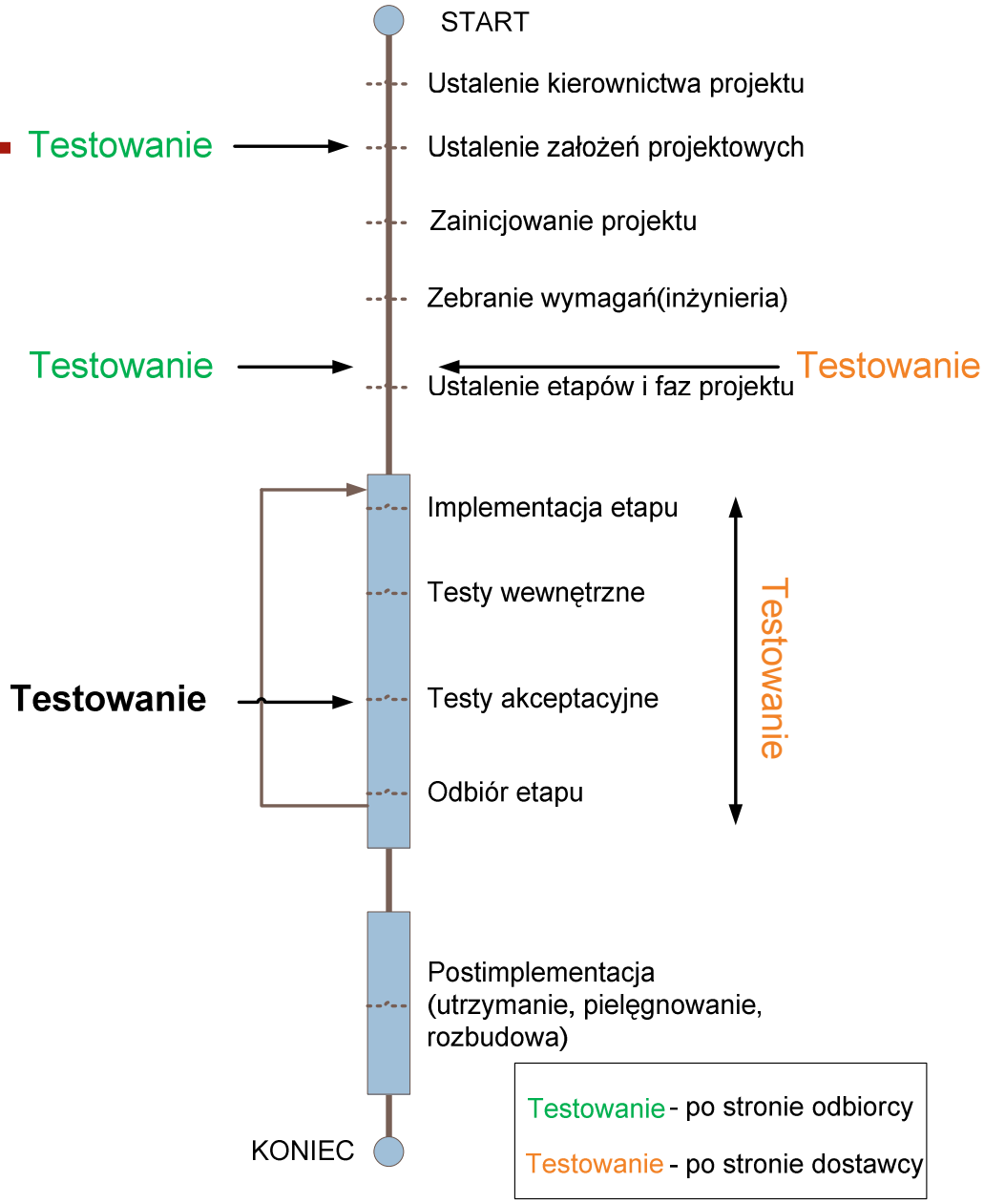
Testowanie – testy akceptacyjne

- Dokument testów akceptacyjnych – **dokładny opis** testów do wykonania, których celem jest:
  - Potwierdzenie wykonania oprogramowania (etapu) odpowiedniej jakości, zgodnie z założeniami projektu.
  - Doprowadzenie do odbioru oprogramowania (etapu).
- Test akceptacyjny powinien zawierać informacje:
  - **Nagłówek**: identyfikator, opis, autor, data utworzenia, dana modyfikacji, wykonawca, data wykonania
  - **Lista przypadków testowych**
  - **Lista scenariuszy testów** dla przypadków testowych
  - **Uwagi**

Dokument testów akceptacyjnych

Linia życia

- Testowanie



Testowanie – Mantis

- **Jedno repozytorium** informacji projektowych
- Dostęp do zadań realizowanych w bieżącym etapie
- Możliwość **zgłaszania uwag** do realizowanych zagadnień
- Interakcja „online” z dostawcą – większa interakcja to **wyższa jakość** projektu, **mniejsze ryzyko**
- Powiadamianie o zdarzeniach drogą emailową

Mantis

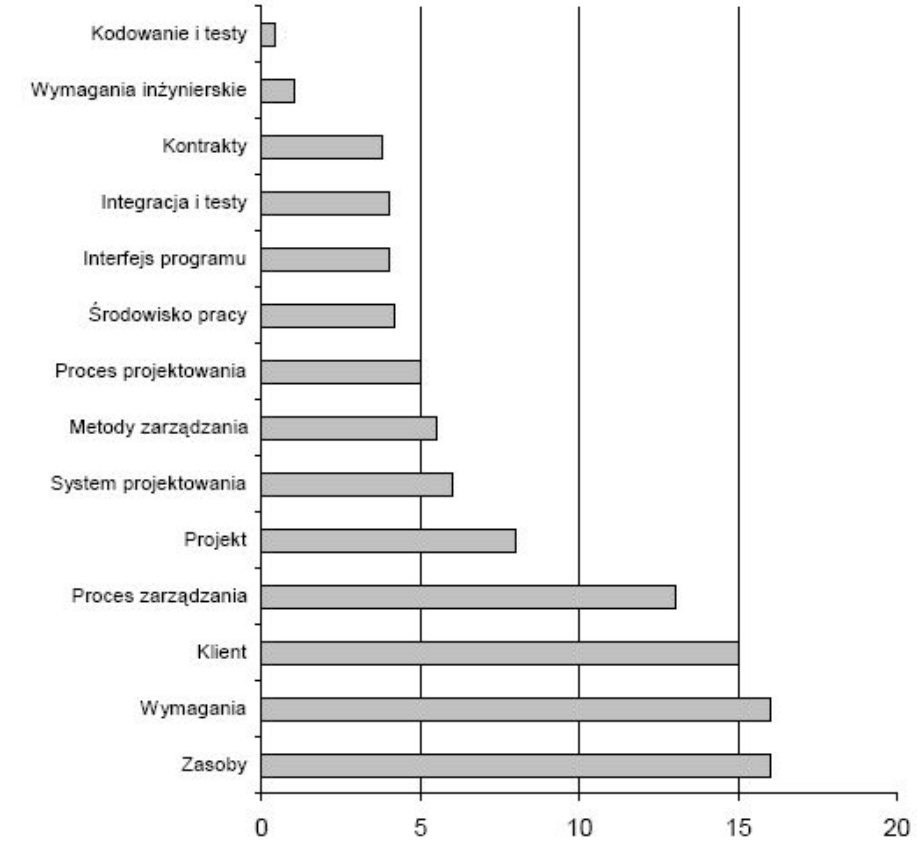
Zarządzanie ryzykiem

- **Ryzyko –** możliwość, prawdopodobieństwo, że coś się nie uda; przedsięwzięcie, którego wynik jest nieznany, niepewny, problematyczny.
- **Zarządzanie ryzykiem:**
  - Świadomość tego, że ryzyko jest **zawsze obecne**
  - Takie „ustawienie” projektu, które **minimalizuje możliwość porażki** wynikającą z aktualizowania się ryzyka
  - Znajdowanie **kompromisu** pomiędzy możliwymi negatywnymi skutkami niesionymi przez ryzyko, a potencjalnymi korzyściami, które daje szansa.

Obszary występowania ryzyka

Źródła ryzyka projektowego, wg SoftwareEngineering Institute

Carnegie Mellon University



Źródła informacji o ryzyku

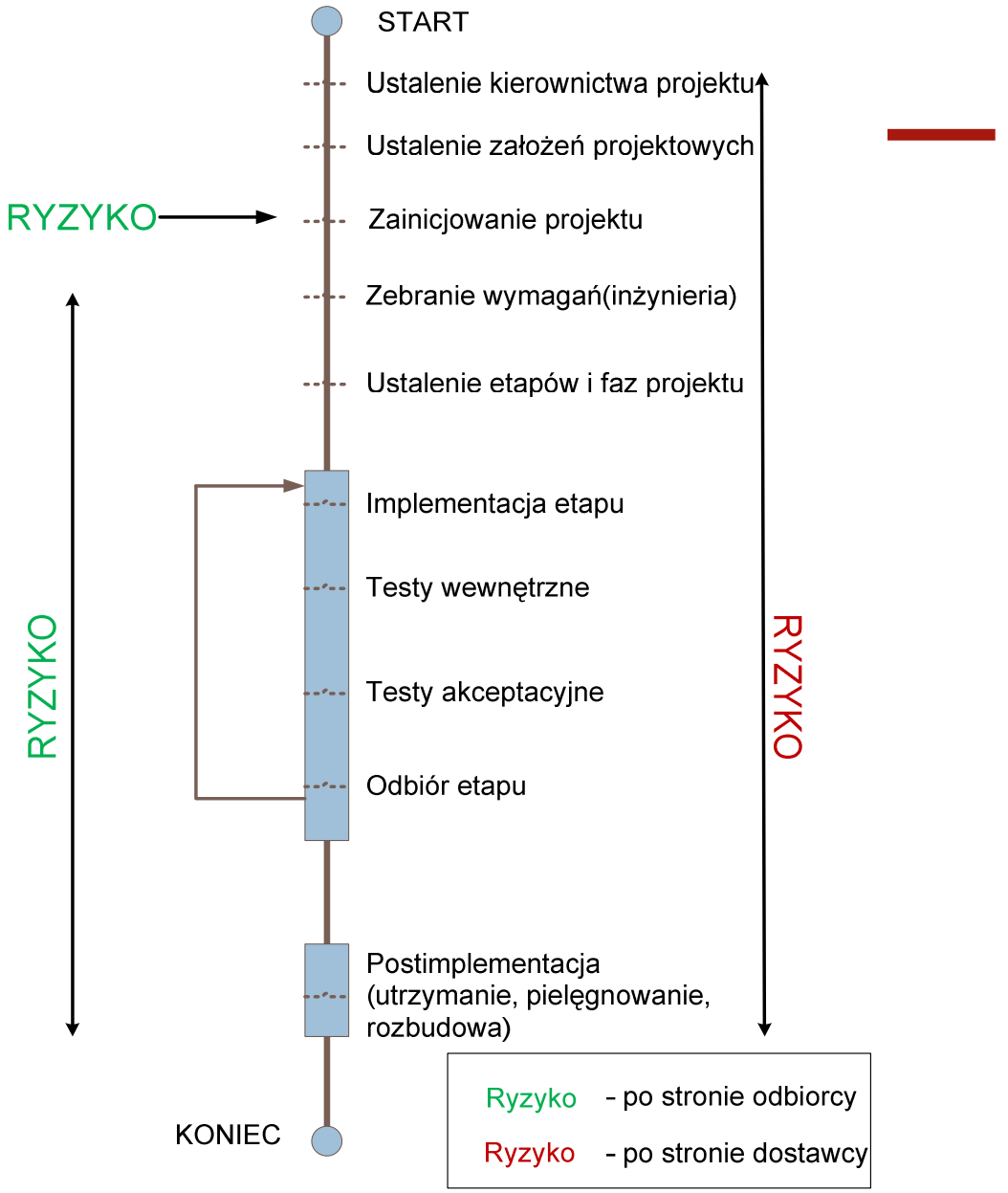
- Zakres projektu i produktu
- Struktura organizacyjna projektu
- Dostępne zasoby
- Harmonogram prac
- Struktura finansowania projektu
- Klient, kooperanci
- Doświadczenie i informacje archiwalne

Narzędzia identyfikacji ryzyka

- Wywiady z udziałowcami projektu
- Audyty projektu i jego procesów
- Kwestionariusze kontrolne
- Zgłoszenia indywidualne (MANTIS)

Linia życia

- Zarządzanie ryzykiem



Rejestr potencjalnego ryzyka projektu

- ***Świadomy potrzeb zarządzania ryzykiem*** dostawca powinien dla projektu prowadzić **rejestr ryzyka** – zbiór formularzy opisujących poszczególne ryzyko.
- **Formularz ryzyka** powinien zawierać:
  - Identyfikator ryzyka
  - Prawdopodobieństwo pojawienia się ryzyka
  - Wartość kosztowa ryzyka – *istotne dla budżetu*
  - Symptomy
  - Opis
  - Wpływ na parametry projektu
  - Sposoby minimalizacji

Wartość kosztowa ryzyka

- Przykład:

Ryzyko nowych wymagań klienta:

10% * 30.000 PLN = 3000 PLN

Ryzyko zmiany kierownictwa projektu:

20% * 50.000 PLN = 10.000 PLN

Macierz szacowania wagi ryzyka

Prawdopodobieństwo **Skutek**	**Bardzo wysokie**	**Wysokie**	**Średnie**	**Niskie**	**Bardzo Niskie**
Katastrofalny	Wysokie	Wysokie	Średnie	Średnie	Niskie
Krytyczny	Wysokie	Wysokie	Średnie	Niskie	Żadne
Marginalny	Średnie	Średnie	Niskie	Żadne	Żadne
Nieistotny	Średnie	Niskie	Niskie	Żadne	Żadne

Zasady zarządzania ryzykiem

- Wspólna wizja produktu
- Praca zespołowa
- Globalna perspektywa
- Myślenie przyszłościowe
- Komunikacja
- Zintegrowane zarządzanie
- Ciągłość procesów

Dziękujemy za uwagę