

Wydział Odlewnictwa

**Wirtualizacja procesów odlewniczych**

Katedra Informatyki Stosowanej WZ AGH



# Cykl życia systemów informatycznych

Projektowanie informatycznych systemów zarządzania produkcją

# Treść wykładu

- ❑ **Cykl życia systemu SDLC**
- ❑ **Metody stosowane w SDLC**
- ❑ **Metodyki alternatywne**

# Cykl życia SI

**System Development Life Cycle (SDLC)** - standardowa metoda rozwijania SI, zakładająca sekwencyjne (ale zachodzące na siebie) wykonanie następujących etapów:

1. analiza istniejącego systemu
2. zdefiniowanie wymagań
3. projektowanie
4. opracowanie nowego systemu lub zakup
5. wdrożenie
6. działanie operacyjne
7. ocena działania systemu
8. utrzymanie i konserwacja

Wynikiem każdego etapu może być zatrzymanie prac lub powrót do któregoś z wcześniejszych etapów.

# Cykl życia SI

## Dlaczego nie zawsze się udaje

- ❑ Użytkownik nie umie wyartykułować swoich potrzeb.
- ❑ Próba wdrożenia nierealnego projektu.
- ❑ Niedopasowanie poszczególnych części systemu.
- ➔ Opracowanie i wdrożenie systemu informatycznego jest dużym przedsięwzięciem organizacyjnym.

# Cykl życia – analiza istniejącego systemu

- ❑ Celem badania jest określenie czy istniejący system spełnia wyznaczone cele i zadania organizacji.
- ❑ Badanie wykonuje specjalnie powołany zespół, którego skład zależy od wielkości organizacji. Zespół jest odpowiedzialny za opracowanie raportu oceniającego potrzebę analizy i zaprojektowania systemu. Decyzję o dalszych pracach podejmuje kierownictwo organizacji.
- ❑ Analiza ma wskazać problemy i ograniczenia istniejącego systemu oraz określić, w jaki sposób rozszerzyć system, by spełniał cele i zadania organizacji:
  - wybór zespołu
  - zebranie danych
  - analiza danych
  - przygotowanie raportu

# Cykl życia – analiza istniejącego systemu

- Należy położyć nacisk na *silne* i *słabe strony* istniejącego systemu przetwarzania danych. Należy przede wszystkim określić:
  - wejścia i wyjścia systemu,
  - kartoteki (zbiory), ich zawartość i sposoby przechowywania,
  - zasady współdziałania użytkowników,
  - metody i procedury przetwarzania danych,
  - działający sprzęt i oprogramowanie.

# Cykl życia – analiza istniejącego systemu

## Zbieranie danych: techniki

- ❑ wywiady (kierowane i swobodne)
- ❑ kwestionariusze
- ❑ przykładowe dokumenty
- ❑ bezpośrednie obserwacje
- ❑ rozmowy telefoniczne
- ❑ testy statystyczne
- ❑ symulacja

# Cykl życia – analiza istniejącego systemu

- **Analiza danych** ma za cel przetworzenie surowych danych w formę dostosowaną do oceny istniejącego systemu. M. in. należy określić:
  - największy, najmniejszy i średni poziom aktywności użytkowników,
  - redundancję procedur,
  - operacje najbardziej pracochłonne,
  - operacje, które wymagają dużych nakładów obliczeniowych,
  - procedury, które stały się zbędne.



# Metody projektowania i analizy systemów

## Podejście systemowe (strukturalne)

- Narzędzia i techniki charakteryzujące się podejściem top-down, w którym użytkownicy analizują system począwszy od dużego stopnia uogólnienia i stopniowo go uszczegółwiają.

# Metody projektowania i analizy systemów

## Podejście systemowe (strukturalne)

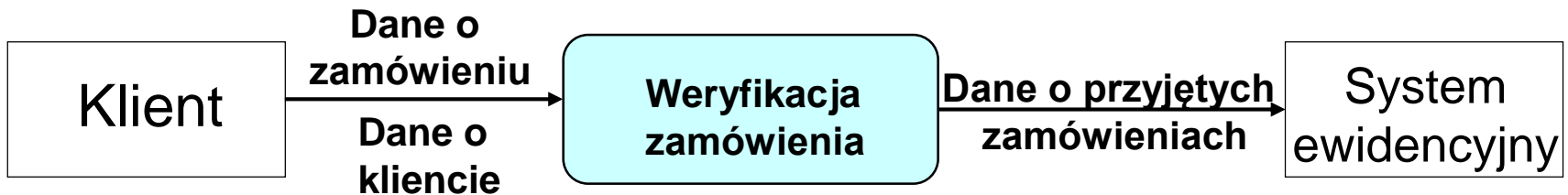
- ❑ Diagram kontekstowy (Context Diagram)
  - Graficznie opisuje ogólną budowę systemu;
- ❑ Diagramy przepływu danych (DFD - Data Flow Diagrams)
  - Przedstawia funkcjonowanie podsystemów,
  - DFD pokazuje 3 czynniki:
    - w jaki sposób dane przepływają przez system (wejścia i wyjścia),
    - procesy przetwarzania danych,
    - gdzie dane są zapamiętywane i przechowywane.
- ❑ Opis danych:
  - diagram związków encji (ERD);
- ❑ Opis procesów:
  - wykresy przepływów w systemie (system flowchart),
  - drzewa decyzyjne,
  - tablice decyzyjne.

# Metody projektowania i analizy systemów

Diagram kontekstowy pokazuje:

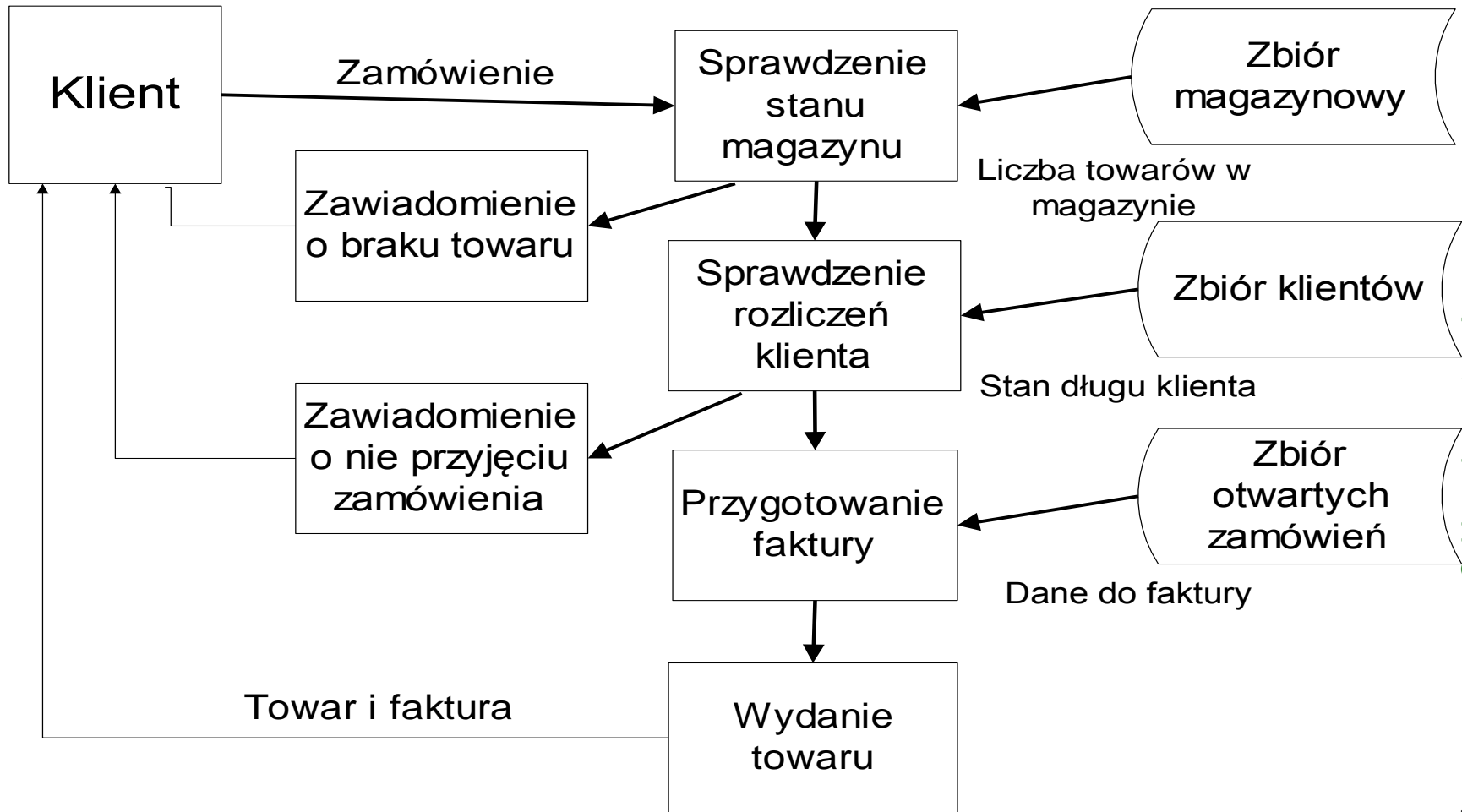
- wejścia, wyjścia i przetwarzanie danych,
- encje (obiekty w systemie).

## System ewidencji zamówień



# Metody projektowania i analizy systemów

## Analiza systemu - diagramy przepływu danych

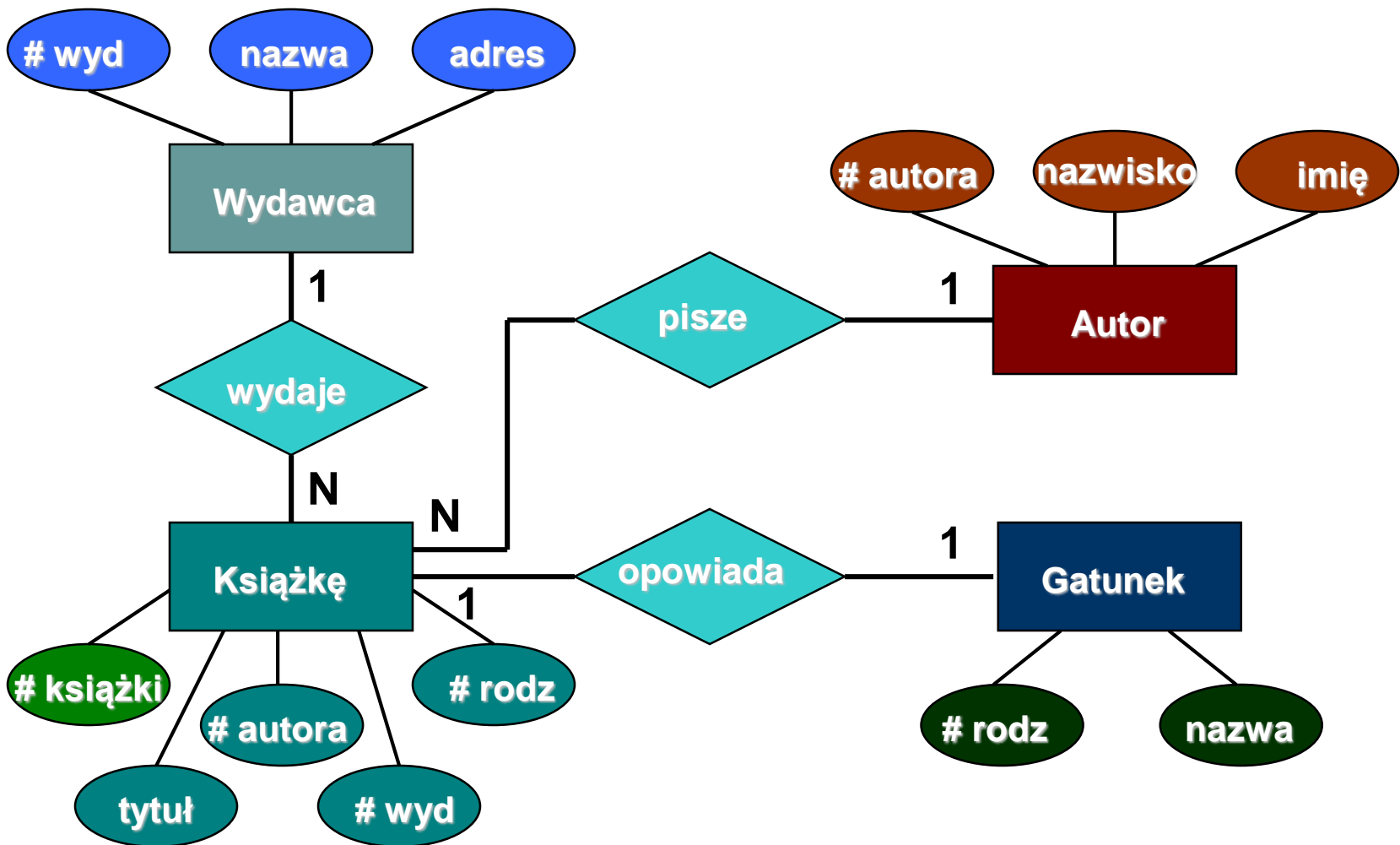


# Metody projektowania i analizy systemów

- **Diagram związków encji** (entity relationship diagram - ERD) jest graficznym opisem, który identyfikuje encje (obiekty) w systemie i relacje między nimi.

# Metody projektowania i analizy systemów

## Analiza danych - diagram związków encji



# Cykl życia – zdefiniowanie wymagań

- ❑ Celem tego etapu jest odpowiedź na pytanie: **co** i **w jaki sposób** będzie robił system. System powinien rozwiązać problemy wykryte w etapie I.
- ❑ Wymagania organizacyjne: wejścia, magazynowanie danych, przetwarzanie i wyjścia.
- ❑ Wymagania wpływające na oprogramowanie i sprzęt.
- ❑ Oszacowanie możliwych wariantów.
- ❑ Opracowanie raportu.

# Cykl życia – projektowanie systemu

- ❑ Ogólny projekt systemu powinien zawierać:
  - uwarunkowania organizacyjne
  - określenie funkcji systemu
  - projekt wyjść
  - projekt procedur przetwarzania
  - projekt wejść
  - projekt zbiorów i baz danych
- ❑ Używane techniki



# Cykl życia – opracowanie nowego systemu

- ❑ Przegląd wymagań oprogramowania pod względem wejść, wyjść i przetwarzania
- ❑ Opracowanie logiki programu
- ❑ Zakodowanie programu
- ❑ Przetestowanie programu
- ❑ Udokumentowanie programu

# Cykl życia – wdrożenie systemu

- ❑ Opracowanie dokumentacji operacyjnej
- ❑ Przeszkolenie użytkowników
- ❑ Przekształcenie zbiorów danych
- ❑ Przetestowanie systemu
- ❑ Praca z nowym systemem

# Cykl życia – wdrożenie systemu

Podejście	Opis
<b>Wdrożenie równoległe</b>	<p>Stary i nowy system funkcjonują równoległe, aż nowy zacznie odpowiednio działać</p> <ul style="list-style-type: none"><li>• Kosztowne, ale bezpieczne</li><li>• Najlepsze dla krytycznych podsystemów</li></ul>
<b>Wdrożenie bezpośrednie</b>	<p>Nowy system natychmiast zastępuje stary</p> <ul style="list-style-type: none"><li>• Mniej kosztowne, ale bardziej ryzykowne</li><li>• Najlepsze dla niekrytycznych podsystemów</li></ul>

# Cykl życia – wdrożenie systemu

Podejście	Opis
<b>Wdrożenie pilotażowe</b>	<p>Jedna z jednostek jest polem doświadczalnym</p> <ul style="list-style-type: none"><li>• Najlepsze dla średniokrytycznych podsystemów</li></ul>
<b>Wdrożenie etapowe</b>	<p>Elementy nowego systemu stopniowo zastępują elementy starego</p> <ul style="list-style-type: none"><li>• Bezpieczne i konserwatywne podejście</li><li>• Najlepsze dla krytycznych podsystemów</li></ul>

# Cykl życia – ocena i utrzymanie systemu

- ❑ Określenie czy system spełnia oczekiwania użytkowników
- ❑ Utrzymanie (konserwacja) systemu:
  - poprawianie błędów,
  - niewielkie zmiany w funkcjonowaniu,
  - regularna aktualizacja.

# Pakiety oprogramowania aplikacyjnego

- ❑ Używane przez firmy, które nie mają doświadczenia lub potrzeby rozwoju systemu jako całości.
- ❑ Dobre dla niekrytycznych aplikacji takich jak: word processing, analizy finansowe, ewidencja magazynowa czy kadrowa.
- ❑ Z reguły pakiety te wymagają dostosowania (ang. **customization**) do specyfiki firmy.

# Pakiety oprogramowania aplikacyjnego

- ❑ Kluczowym elementem sukcesu w tej metodzie jest dokładne sprecyzowanie wymagań odnośnie cech, jakie musi spełniać oferowane oprogramowanie. Porównanie różnych ofert pozwoli wybrać to, które w największym stopniu spełnia nasze wymagania.

# Systemy tworzone przez użytkowników

- ❑ Rozwój systemów informatycznych przez użytkowników końcowych z małym lub nawet bez formalnego wsparcia specjalistów-informatyków.
- ❑ Pozwalają użytkownikom na uwzględnienie swoich specyficznych potrzeb biznesowych.
- ❑ Podejście właściwe w przypadku małych podsystemów, nie przetwarzających masowych transakcji.



# Outsourcing

- ❑ Praktyka polegająca na zleceniu obsługi informatycznej i telekomunikacyjnej do usługodawców zewnętrznych.
- ❑ Dokonywane ze względów ekonomicznych lub technicznych.

<b>Funkcje</b>	<b>Udział</b>
<b>Dzierżawa PC i sieci</b>	<b>35%</b>
<b>Wsparcie techniczne</b>	<b>34%</b>
<b>Rozwój aplikacji</b>	<b>30%</b>
<b>Brak</b>	<b>24%</b>
<b>Obsługa systemu</b>	<b>22%</b>
<b>Konserwacja systemu</b>	<b>19%</b>
<b>Inne</b>	<b>19%</b>
<b>Call center</b>	<b>15%</b>

# Cloud computing

- ❑ Salesforce.com - to aplikacje udostępniane przez Internet (cloud), a nie tradycyjne rozwiązania wymagające zakupu i instalacji oprogramowania z pełną licencją.
- ❑ IBM i Google - to setki połączonych komputerów i serwerów z wielordzeniowymi procesorami, dzięki którym można udostępnić moc obliczeniową innym firmom.
- ❑ Amazon.com - udostępnia w ramach tego typu usług całą infrastrukturę IT, której nie wykorzystuje w pełni na swoje potrzeby.
- ❑ Rozwiązania są znane – właściwie chodzi o nowy model biznesowy.

# Cloud computing

- ❑ W odróżnieniu od tradycyjnych rozwiązań, w których każdy użytkownik ma komputer, zestaw programów na twardym dysku i dzieli się wynikami pracy za pomocą emaili, w CC oprogramowanie jest instalowane tylko na jednym serwerze. Dzięki połączeniu z "jednostką centralną" użytkownicy mogą pracować w nawet bardzo wyrafinowanych programach bez potrzeby instalowania czegokolwiek na swoich maszynach. Dodatkowo zyskują możliwość łatwego dzielenie się owocami pracy, publikowania ich, a nawet udostępnianie ich do edycji innym osobom.
- ❑ W skład Google Apps wchodzi Dokumenty Google, czyli zestaw **darmowych** stron, zawierających aplikacje podobne do Microsoft Office. Zawierają one edytor tekstu, arkusz kalkulacyjny i program do tworzenia prezentacji. Ich uzupełnienie stanowi Picasa oraz zintegrowane z całym zestawem poczta Gmail i serwis blogowy Blogger umożliwiające łatwe dzielenie się plikami oraz publikowanie ich w sieci na prywatnym blogu. Ponadto dzięki Google Gears, dodatkowi do systemów IE i Firefox, pracować na nich można nawet bez połączenia z siecią. Dokumenty możemy zapisać na twardym dysku, a po nawiązaniu łączności wersja internetowa zostanie automatycznie uzupełniona.
- ❑ Google – Google Docs, [docs.google.com](https://docs.google.com)

# Rozwój systemów – czynniki sukcesu

- ❑ Wszystkie systemy informatyczne są systemami biznesowymi
  - Wiele projektów kończy się niepowodzeniem, bo są postrzegane jako projekty techniczne, a nie biznesowe.
- ❑ Reagowanie na zmiany
  - W miarę zmian w otoczeniu systemu, system musi się zmieniać.
- ❑ Rozwój systemu jest procesem ryzykownym

# Rozwój systemów – czynniki sukcesu

- ❑ Standardowe rozwiązania nie są wystarczające – szczególnie w globalnym otoczeniu
  - Należy uwzględniać uwarunkowania techniczne, kulturowe i polityczne.
- ❑ Zespół ds. rozwoju
  - IS menadżer musi zbudować zespół zdolny do twórczej, innowacyjnej i grupowej pracy.
- ❑ Użytkownicy
  - Kluczowym czynnikiem sukcesu nie jest sprzęt ani oprogramowanie, lecz użytkownicy.