


Inżynieria Oprogramowania

# Wprowadzenie



## Tematy

- Profesjonalne wytwarzanie oprogramowania
  - Co to jest inżynieria oprogramowania
- Często zadawane pytania (FAQ) nt. inżynierii oprogramowania

9

## Słownik

- Development
  - Rozwój, produkcja, wytwarzanie

10

## Inżynieria Oprogramowania

- Ekonomia wszystkich krajów rozwijających się opiera się na wykorzystaniu oprogramowania.
- Coraz większa liczba systemów jest kontrolowana za pomocą oprogramowania.
- Systemy oprogramowania są abstrakcyjne i nieuchwytne.
- Inżynieria Oprogramowania związana jest z teoriami, metodami i narzędziami do profesjonalnego wytwarzania oprogramowania.

## Koszty oprogramowania



- Koszty oprogramowania są często dominujące – przewyższają koszty sprzętu.
- Koszty wsparcia oprogramowania są wyższe niż koszty jego produkcji
  - Dla systemów długoterminowych mogą być wielokrotnie wyższe.
- Efektywne wytwarzania oprogramowania, jest istotą Inżynierii Oprogramowania.

## Oprogramowanie jako produkt



- **Oprogramowanie generyczne/ogólne** (ang. generic)
  - Odrębne systemy wytwarzane i sprzedawane szerokiemu zakresowi klientów.
  - Przykłady – Programy graficzne, oprogramowanie wspierające zarządzanie projektami, systemy CAD, systemy specyficzne dla określonej dziedziny np. system obsługi apteki.
- **Oprogramowanie dopasowane** (ang. custom)
  - wyprodukowane dla konkretnego klienta zgodnie z dostarczoną specyfikacją
  - Przykłady – systemy wbudowane, systemy kontroli ruchu lotniczego, systemy monitoringu i sterowania ruchem ulicznym.

## Specyfikacja produktu



- Oprogramowanie generyczne
  - Specyfikacja tego, jak oprogramowanie powinno działać, jest własnością **wykonawcy**,
  - decyzje związane ze zmianami podejmuje **wykonawca**.
- Oprogramowanie dopasowane
  - Specyfikacja tego, jak oprogramowanie powinno działać, jest własnością **klienta**,
  - decyzje związane ze zmianami podejmuje **klient**.

14

## Model dostarczania



1. Instalacja po stronie klienta
  - System uruchomiony na infrastrukturze klienta (własnej lub dzierżawionej)
  - Klient -> właściciel
2. Udostępnianie zdalne
  - System uruchomiony na infrastrukturze dostawcy
  - Oprogramowanie jako usługa (ang. Software as a Service SaaS)
  - Dostawca -> właściciel, Klient -> Dzierżawca

15

## Atrybuty dobrego oprogramowania



- Oprogramowanie powinno dostarczać użytkownikom wymaganą funkcjonalność i wydajność.
- Oprogramowanie powinna cechować:
  1. **Zarządzalność**: musi ewoluować w odpowiedzi na zmieniające się wymagania
  2. **Wiarygodność i bezpieczeństwo**: można na nim polegać
  3. **Sprawność**: nie powinno marnować zasobów systemu
  4. **Użyteczność**: musi być użyteczne dla użytkowników, dla których zostało zaprojektowane

## Inżynieria Oprogramowania



- Inżynieria Oprogramowania to **dyscyplina inżynierska** dotycząca **wszystkich aspektów związanych z wytwarzaniem oprogramowania**
  - poczynając od wstępnych zarysów specyfikacji
  - kończąc na wspieraniu systemu w czasie jego działania.
- Dyscyplina inżynierska
  - Wykorzystanie adekwatnych teorii i metod do rozwiązywania problemów z uwzględnieniem ograniczeń organizacyjnych i finansowych.
- Wszystkie aspekty związane z wytwarzaniem
  - Nie tylko techniczny proces produkcji ale również zarządzanie projektem oraz rozwój narzędzi, metod itp. Wspierających rozwój oprogramowania.

## Znaczenie inżynierii oprogramowania



- Coraz więcej jednostek i całych społeczeństw polega na rozwoju systemów oprogramowania. Musimy być zdolni do **szybkiego i ekonomicznego** wytwarzania **wiarygodnych i niezawodnych** systemów.
- W szerokim horyzoncie czasowym, wykorzystanie **metod i technik Inżynierii Oprogramowania** (projekt zespołowy), w miejsce po prostu **pisania programów** (projekt indywidualny), jest po prostu bardziej opłacalne.
  - Dla większości typów systemów najwyższe koszty umiejscowione są w obszarze wprowadzania zmian w działającym oprogramowaniu.

21


## Czynności procesu Inżynierii Oprogramowania



- Specyfikacja oprogramowania
  - Klient i inżynierowie definiują oprogramowanie, które ma zostać wyprodukowane oraz określają ograniczenia przy których ma realizować swoje zadania.
- Rozwój (produkcja) oprogramowania
  - Oprogramowanie jest projektowane i implementowane.
- Zatwierdzanie (walidacja) oprogramowania
  - oprogramowanie jest sprawdzane pod kątem spełniania wymagań klienta.
- Ewolucja oprogramowania
  - Oprogramowanie jest modyfikowane w celu uwzględnienia zmian w wymaganiach klienta oraz środowiska (np. rynkowego, prawnego, sprzętowego, ...).

22


## Ogólne problemy mające wpływ na oprogramowanie



- Heterogeniczność
  - Coraz częściej systemy muszą działać w środowiskach rozproszonych, które obejmują różne rodzaje systemów komputerowych i urządzeń mobilnych.
- Zmiany środowiska biznesowego i społecznego
  - Biznes i społeczeństwo podlegają szybkim zmianom wymuszonym przez rozwój gospodarczy i dostępność nowych technologii. Powoduje to potrzebę wprowadzania zmian w istniejącym oraz szybką produkcję nowego oprogramowania.
- Bezpieczeństwo i zaufanie
  - Ponieważ oprogramowanie splata się z niemal wszystkimi aspektami naszego życia bardzo istotne jest to czy możemy mu ufać.

23

## Inżynieria oprogramowania a inne dyscypliny



- **Informatyka** (ang. computer science) związana jest z podstawami teoretycznymi; inżynieria oprogramowania związana jest z praktyką budowania i dostarczania użytecznego oprogramowania.
- **Inżynieria systemów** (ang. system engineering) jest związana z wszelkimi aspektami budowy i ewolucji złożonych systemów
  - w których oprogramowanie może grać kluczową rolę

24

## Niejednolitość Inżynierii Oprogramowania



- Istnieje wiele różnych typów oprogramowania
  - Nie istnieje zestaw uniwersalnych technik i narzędzi, który można by zastosować w produkcji.
- Metody i narzędzia wykorzystywane do produkcji oprogramowania będą się różnić w zależności od:
  - Typu rozwijanej aplikacji
  - Wymagań klienta
  - Umiejętności, wiedzy i doświadczenia zespołu projektowego.

25

## Typy aplikacji



- Aplikacje samodzielne (lokalne)
  - Aplikacje działające na lokalnym komputerze i posiadające wszystkie wymagane funkcjonalności bez potrzeby łączyć się z siecią.
- Interaktywne aplikacje transakcyjne
  - Aplikacje wykonywane na zdalnym komputerze, do których użytkownicy mają dostęp za pośrednictwem własnego komputera. (aplikacje webowe np. e-commerce).
- Wbudowane systemu kontrolne
  - Oprogramowanie kontrolujące i zarządzające działaniem urządzeń. W ujęciu ilościowym tego typu systemów jest prawdopodobnie więcej niż wszystkich innych typów razem wziętych.

26



## Typy aplikacji

- Systemy przetwarzania wsadowego
  - Systemy biznesowe przeznaczone do przetwarzania dużych ilości danych (wsadów). Przetwarzają duży zestaw pojedynczych danych wejściowych w celu utworzenia odpowiadającego im rezultatu.
- Systemy rozrywkowe
  - Systemy, przede wszystkim do użytku osobistego, przeznaczone do rozrywki.
- Systemy do modelowania i symulacji
  - Systemy opracowywane przez naukowców i inżynierów dla celów modelowania fizycznych procesów lub sytuacji, które obejmują wiele odrębnych, ale będących ze sobą w interakcji, obiektów.

27

## Typy aplikacji

- Systemy gromadzenia danych
  - Systemy zbierające, za pośrednictwem czujników, dane ze środowiska i przesyłające je do innych systemów dla celów dalszego przetwarzania.
- Systemy systemów
  - Systemy złożone z wielu innych systemów.

28

## Fundamenty inżynierii oprogramowania



- Niezależnie od typu oprogramowania i techniki jego opracowania istnieje zbiór podstawowych zasad, które należy stosować:
  - System powinien być wytwarzany przy użyciu zarządzalnego i zrozumiałego procesu (inżynierii oprogramowania). Oczywiście do różnych typów systemów stosuje się różne procesy.
  - niezawodność i wydajność są ważne.
  - Zrozumienie wymagań i zarządzanie specyfikacją oprogramowania są ważne.
  - Tam gdzie jest to właściwe, zamiast budowania nowego, powinno się powtórnie wykorzystywać oprogramowanie, które zostało już opracowane.

29

## Inżynieria oprogramowania a Internet



- Internet (Web) jest obecnie platformą wykonawczą aplikacji. Organizacje coraz intensywniej angażują się w rozwój systemów bazujących na technologiach internetowych (Webowych) kosztem systemów lokalnych.
- Usługi Internetowe (ang. Web Services) pozwalają na dostęp do funkcjonalności za pośrednictwem Internetu.
- Chmury obliczeniowe (ang. Cloud computing) to podejście do dostarczania usług komputerowych w którym aplikacje wykonywane są zdalnie w „chmurze”.
  - Użytkownik nie kupuje oprogramowania tylko płaci za jego użycie.

30

## Inżynieria systemów internetowych



- Internet doprowadził do znaczącej zmiany w sposobie organizacji oprogramowania biznesowego.
- Dominującym podejściem w konstrukcji oprogramowania webowego jest wielokrotne użycie.
  - Budując tego typu systemy skupiamy się na wymyśleniu w jaki sposób można złożyć system z istniejących komponentów oraz systemów.
- Systemy webowe powinny być rozwijane i dostarczane przyrostowo.
  - Obecnie powszechnie uznaje się, że nie jest możliwe z góry określenie wszystkich wymagań dla takich systemów.
- Interfejsy użytkownika są ograniczane możliwościami przeglądarek Internetowych.
  - Technologie takie jak AJAX pozwalają na realizację rozbudowanych i szybkich interfejsów w ramach przeglądarki. Na szeroką skalę wykorzystywane są formularze webowe oraz skrypty wykonywane lokalnie.
  - Obecnie systemy internetowe zaczynają być konkurencyjne z systemami PC w zakresie dziedzinie interfejsów użytkownika.

31

## Wyzwania w dużych projektach



- Zbudowanie skomplikowanego systemu jest dużym wyzwaniem
  - Duży nakład pracy
  - Wysokie koszty
  - Długi czas wytworzenia
  - Zmieniające się potrzeby użytkowników
  - Ryzyko:
    - Porażki projektu, braku akceptacji przez użytkowników, małej wydajności, trudnej konserwacji, ...
- (Inaczej niż w przypadku jednorazowych programów, gdzie często twórca jest użytkownikiem)

## Sukces projektu oprogramowania



- Projekty, mające na celu opracowanie oprogramowania nie zawsze kończą się sukcesem
- Kiedy możemy uznać, że oprogramowanie jest sukcesem?
  - Projekt został ukończony
  - Oprogramowanie jest użyteczne
  - Oprogramowanie jest „używalne”
  - Oprogramowanie jest używane
  - Oprogramowanie jest opłacalne i zarządzalne

## Przyczyny porażki



- Poślizg czasowy
- Przekroczenie zakładanego budżetu
- Nie rozwiązywanie problemów użytkowników
- Słaba jakość
- Problemy z zarządzaniem

## Przyczyny porażki...



- Do tych problemów prowadzi najczęściej doraźność projektów software'owych
  - Brak planowania prac
  - Brak zdefiniowania tego co ma zostać zrealizowane
  - Nie zrozumienie wymagań użytkowników
  - Brak kontroli i przeglądów
  - Niekompetencja
  - Nie zdawanie sobie sprawy z rzeczywistych kosztów i potrzebnego nakładu pracy (przez twórców i użytkowników)

## Inne dyscypliny



- Wielkie projekty nie są niczym nadzwyczajnym ... tak jak i ich sukcesy
  - Mosty, zapory, stadiony
  - Elektrownie
  - Samoloty, satelity
- Zwykle rozwiązaniem jest „inżynieria”:
  - projektowanie i konstruowanie obiektów oraz urządzeń technicznych
    - efektywnie spełniających wymagania z punktu widzenia kosztów, jakości, wydajności,...

## Inżynieria



- Wymaga dobrze zdefiniowanego podejścia: powtarzalnego, przewidywalnego
- Duże projekty wymagają zarządzania samym projektem
  - Zarządzanie zasobami (ludzkimi, wyposażeniem), budżetem, terminarzem
  - Skala robi ogromną różnicę (porównaj: budowanie baraku, dwupiętrowego domu, 50 piętrowego drapacza chmur).
- Jakość jest wartością samą w sobie (uszkodzenia, efektywność, użyteczność)
  - Ludzie naprawdę chcą płacić za jakość!

## Duże projekty



- Angażują różnych specjalistów
  - Budownictwo: architekt, projektant, inżynier elektryk, robotnicy (murarz, stolarz, hydraulik,...), ...
- Stały nadzór w celu zapewnienia jakości
  - Kontrola jakości cementu, stali, zapewnianie odpowiedniej proporcji piasku i cementu w zaprawie, ...

## Duże projekty ...



- Wiele rezultatów:
  - plan architektoniczny, model, diagram struktury, układ okablowania elektrycznego, ...
- Normy, przepisy, konwencje, które powinny być przestrzegane
- Zdefiniowane kroki, etapy, wykonywane przeglądy, postęp jest widoczny

## Projekty software'owe



- Oprogramowanie różni się od innych produktów
  - Koszty koncentrują się na rozwoju
  - Konserwacja to poprawianie błędów i udoskonalanie oraz dodawanie nowych funkcjonalności
  - Postęp projektu trudno jest mierzyć

## Podsumowanie



- Inżynieria Oprogramowania to dyscyplina inżynierii związana z wszystkim aspektami produkcji oprogramowania.
- Najważniejsze cechy produktu będącego oprogramowaniem to zdolność do konserwacji, niezawodność i bezpieczeństwo, efektywność oraz akceptowalność.
- Podstawowe czynności każdego procesu Inżynierii Oprogramowania to specyfikacja, rozwój, zatwierdzanie oraz ewolucja.
- Istnieje wiele różnych typów systemu, a każdy wymaga odpowiednich narzędzi Inżynierii Oprogramowania oraz techniki rozwoju.
- Podstawowe pojęcia Inżynierii Oprogramowania są powszechnie stosowane do rozwoju wszystkich typów systemów.