

Wprowadzenie

Głównym celem zajęć jest sprawdzenie i poprawa praktycznych umiejętności w zakresie wybranych istotnych aspektów realizacji projektów informatycznych, t.j.:

- Zarządzania wymaganiami
- Planowania prac
- Wyboru architektury, stosowanych technologii
- Wykorzystania narzędzi informatycznych

W celu bardziej realistycznego odwzorowania rzeczywistego procesu produkcji oprogramowania, wskazane jest aby zespoły realizujące każdy z projektów składały się z 3-4 osób. W ramach każdego zespołu osoby powinny się podzielić rolami, przejmując odpowiedzialność za poszczególne aspekty procesu. Choć wybór ról i zakresu odpowiedzialności może być (do pewnego stopnia) dowolnie kształtowany w ramach zespołu, jako przykład można wskazać następujące role:

- a) Architekta
Czyli osoby która projektuje konstrukcję programu, proponuje/wybiera rozwiązania, technologie itp. Odpowiada również za podział na moduły i ew. definicję interfejsów komunikacyjnych
- b) Testera
Osoby odpowiedzialnej za przygotowanie testów automatycznych, testowanie i raportowanie błędów oraz kontrolę jakości
- c) Analityka / zarządzającego wymaganiami
Osoba która odpowiada za zbieranie wymagań klienta oraz przełożenie ich na wymagania i konstrukcje programistyczne
- d) Kontrolera
Odpowiedzialnego za pilnowanie harmonogramu, zebranie wszystkich elementów projektu w całość itp.
- e) Programisty
Ta rola nie wymaga zapewne komentarza =)

Jedna osoba może (i w wielu przypadkach będzie musiała) pełnić więcej niż jedną rolę. Zespół powinien jednak podzielić się rolami już na początku prac. Przy ocenie projektu będzie brane pod uwagę przede wszystkim inżynierskie podejście – odpowiedni projekt, szacowania (nakładu pracy, ryzyka itp.), analiza możliwych do wykorzystania technologii, ich wad i zalet itp. Samo osiągnięcie założonej funkcjonalności nie jest wystarczające.

Oceniana będzie:

- Prawidłowość zbierania i zarządzania wymaganiami – kompletność zidentyfikowanych wymagań, zarządzanie zmianami itp.
- Projekt realizowanego systemu / aplikacji – schemat/model funkcjonalny, struktury danych, stos technologiczny (krytyczna analiza technologii)
- Jakość projektu: ilość błędów, stopień pokrycia testami automatycznymi, dokumentacja itp.
- Stopień realizacji założonej funkcjonalności

Ocena odbywać się będzie na każdym z etapów prezentacji projektu (patrz następna strona)

Etapy prac i kluczowe terminy

Etap: Wybór tematu (do 18.X)

Studenci mogą zaproponować realizację własnego tematu (aplikacji / systemu itp.) pod warunkiem jego odpowiedniej złożoności. Jako przykłady mogą posłużyć:

- Bank bitcoinów
- Sieciowa gra w brydża
- Platforma studenckiej wymiany książek
- itp.

W zależności od ilości zgłoszonych / uzgodnionych projektów przygotuję listę tematów i przydzielone zostaną dla grup które nie miały własnego pomysłu.

Etap: Projekt systemu (15.XI)

Projekt systemu powinien zawierać zarówno elementy związane z przygotowywanym systemem / aplikacją jak i metodyką prowadzenia projektu, to jest:

- Listę zebranych wymagań
- Architekturę systemu
- Opis interfejsów
- Listę wykorzystywanych technologii (stos technologiczny)
- Projekt testów
- Listę narzędzi używanych przy realizacji projektu (issue tracking, repozytorium itp.)

Ocena z projektu to 30% oceny z przedmiotu

Etap: Prezentacja rezultatów (13.XII)

Na tym etapie zespół ma za zadanie przedstawić funkcjonalny prototyp systemu/aplikacji, obejmujący ok. 50-75% całej funkcjonalności. Należy również przygotować szkielet testów (pokrycie testami na poziomie ponad 30%) oraz szkielet dokumentacji. W czasie prezentacji grupa powinna przedstawić działającą aplikację (w zakresie zaimplementowanej funkcjonalności). Na tym etapie możliwa będzie też zmiana / modyfikacja wymagań

Ocena z tej prezentacji to 30% oceny z przedmiotu

Etap: Dostarczenie kompletnego projektu (17.I)

Na ostatnim etapie zespół ma przedstawić kompletny i funkcjonalny system/aplikację. Należy też poddać aplikację statycznej analizie kodu (przy pomocy np. SonarQube, PMD, FindBugs lub podobnych). Zespół powinien zaprezentować działającą aplikację oraz wnioski z wybranego sposobu tworzenia aplikacji – zidentyfikowane problemy, kluczowe decyzje i ich wpływ na projekt itp.

Istotne elementy oceny programu / aplikacji:

1. Funkcjonalność aplikacji
2. Jakość aplikacji (ilość błędów)
3. Zgodność ze wszystkimi wymaganiami (np. licencyjnymi)
4. Kompletność, czytelność i użyteczność dokumentacji (włączając w to ew. dokumentacje w kodzie)

5. Ergonomia instalacji i użytkowania
6. Zakres dołączonych testów automatycznych

Wymagania dla kodu źródłowego

II. Wykaz elementów składających się na oddawany projekt:

1. Kod źródłowy (wyczyszczony – bez zbędnych elementów typu pliki kontroli wersji itp.)
2. Instrukcje kompilacji, instalacji i konfiguracji
3. Dokumentacja użytkownika (czyli opis interfejsu - co gdzie i jak można zrobić)
4. Dokumentacje techniczna - podział aplikacji na moduły, architektura, opis API i istotnych interfejsów itp.
5. Dokumentacja w kodzie – komentarze
6. Testy automatyczne (jednostkowe i funkcjonalne) aplikacji

III. Wymagania licencyjne / prawne:

1. Kod przygotowany przez grupę w ramach projektu dostarczony na licencji zgodnej z **BSD** (informacje w nagłówkach)
2. Wszystkie wykorzystane biblioteki / komponenty itp. - na licencji pozwalającej na nieograniczone kopiowanie i rozpowszechnianie

Ocena z tego etapu to 40% oceny z przedmiotu