

## Projekt z Inżynierii Oprogramowania

### Skład grupy:

236689  
236807  
236631  
230331  
236727  
236543

### Terminy zajęć projektowych

1. Zajęcia 2019-02-28	14:15	15:00	Dr inż. Zofia Kruczkiewicz	015	D-1
2. Zajęcia 2019-03-07	13:15	15:00	Dr inż. Zofia Kruczkiewicz	015	D-1
3. Zajęcia 2019-03-21	13:15	15:00	Dr inż. Zofia Kruczkiewicz	015	D-1
4. Zajęcia 2019-04-04	13:15	15:00	Dr inż. Zofia Kruczkiewicz	015	D-1
5. Zajęcia 2019-04-15	13:15	15:00	Dr inż. Zofia Kruczkiewicz	015	D-1 pn
6. Zajęcia 2019-04-25	13:15	15:00	Dr inż. Zofia Kruczkiewicz	015	D-1
7. Zajęcia 2019-05-16	13:15	15:00	Dr inż. Zofia Kruczkiewicz	015	D-1
8. Zajęcia 2019-05-30	13:15	15:00	Dr inż. Zofia Kruczkiewicz	015	D-1

### Terminy zajęć laboratoryjnych

1. Zajęcia 2019-02-25	07:30	09:00	Dr inż. Zofia Kruczkiewicz	015	D-1
2. Zajęcia 2019-03-04	07:30	09:00	Dr inż. Zofia Kruczkiewicz	015	D-1
<b>3. Zajęcia 2019-03-11</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
<b>4. Zajęcia 2019-03-18</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
<b>5. Zajęcia 2019-03-25</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
<b>6. Zajęcia 2019-04-01</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
<b>7. Zajęcia 2019-04-08</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
<b>8. Zajęcia 2019-04-29</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
<b>9. Zajęcia 2019-05-06</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
<b>10. Zajęcia 2019-05-13</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
<b>11. Zajęcia 2019-05-20</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
<b>12. Zajęcia 2019-05-27</b>	<b>07:30</b>	<b>09:00</b>	<b>Dr inż. Zofia Kruczkiewicz</b>	<b>015</b>	<b>D-1</b>
13. Zajęcia 2019-06-03	07:30	09:00	Dr inż. Zofia Kruczkiewicz	015	D-1
14. Zajęcia 2019-06-10	07:30	09:00	Dr inż. Zofia Kruczkiewicz	015	D-1
15. Zajęcia 2019-06-17	07:30	09:00	Dr inż. Zofia Kruczkiewicz	015	D-1

Tabela 1. Realizowany projekt

Grupa studencka	Projekt	Zespoły wykonawców	Podgrupy w zespołach
	<b>Wypożyczalnia kaset VHS oraz płyt DVD</b>	Liczba osób <b>6</b>	<ul style="list-style-type: none"> <li>• Jeden student w roli *Scrum Master i Administrator</li> <li>• 5 studentów tworzy **Development Team, w tym jeden ze studentów pełni rolę inspektora, a drugi kierownika</li> </ul>

Tabela 2. Przebieg realizacji każdego z projektów (tabela 1)

Opis realizacji projektu					
Nr tygodni Daty	Termin przekazania prac	Spotkanie	Uwagi dotyczące realizacji zadań przez każdą z dwóch podgrup zespołu	Ocena częstkowa	Zadania Scrum Master
<b>28.02.19</b> 1-tydzień	Wynik prac należy wysłać za pomocą poczty elektronicznej do dnia <b>19.03.19</b> Wynik prac zostanie oceniony przez prowadzącego zajęcia.	Sprint planning meeting (45 min)	1) Zajęcia organizacyjne ( podział na grupy i podgrupy, przydzielenie ról projektowych, uzyskanie dostępu do wymaganych narzędzi) 2) User Stories - Opracowanie modelu biznesowego „świata rzeczywistego” systemu		Współdziałanie z wykonawcami
<b>7.03.19</b> 2- tydzień 3- tydzień	<b>Uwaga:</b> należy wysłać wersję końcową projektu UML z ewentualnie uzupełnionym diagramem wymagań	Sprint planning meeting (45 min)	<ul style="list-style-type: none"> <li>• Sprint Backlog (formy pośrednie: Product Backlog), Sprint planining) - zdefiniowanie wymagań funkcjonalnych i нефункциональных wynikających z otrzymanego fragmentu opisu „świata rzeczywistego”</li> </ul> <p>Podczas spotkania 7.03.2018 należy zaprezentować wyniki prac, czyli „Opis świata rzeczywistego” i oraz listę wymagań funkcjonalnych i нефункциональных.</p> <p><b>Należy wykonać:</b></p> <p>1) na podstawie listy wymagań funkcjonalnych i нефункциональных diagram wymagań wg instrukcji: <b>Instrukcja_3 - definiowanie wymagań</b></p> <p>2) diagram przypadków użycia wg instrukcji do lab4: <b>Instrukcja_4 - specyfikacja wymagań za pomocą diagramu przypadków użycia.</b></p> <p>ze specyfikacją poszczególnych przypadków użycia</p>		<b>236543</b>  Udział w pracach projektowych

<p><b>21.03.19</b> 4- tydzień 5- tydzień</p>	<p>Projekt UML oraz projekt z kodem Javy należy wysłać za pomocą poczty elektronicznej do dnia <b>2.04.19</b></p>	<p>Sprint planning meeting (45 min)</p>	<p>3) Diagramy klas i sekwencji wg instrukcji do lab5-7: <b>Instrukcja - część 1</b> <b>Instrukcja - część 2</b> Diagram klas UML został przedstawiony na wykładzie w dniu 25.03.2019.</p>		<p>Współdziałanie z wykonawcami  Eliminacja redundancji w projekcie. Udział w pracach projektowych</p>
<p><b>4.04.19</b> 6- tydzień 7-tydzień (4 dni)</p>	<p>Projekt UML oraz projekt z kodem Javy należy wysłać za pomocą poczty elektronicznej do dnia <b>13.04.19</b></p>	<p>Sprint planning meeting (45 min)</p>	<p>4) Projektowanie i implementacja oprogramowania w oparciu o wzorce projektowe – diagramy UML: rozwijanie diagramu klas, diagramy sekwencji i aktywności wybranych 3 prostych przypadków użycia reprezentujących projekt warstwy biznesowej projektu oraz implementacja projektu na platformie Java SE. Diagramy klas i sekwencji wg instrukcji do lab5-7: <b>Instrukcja - część 1</b> <b>Instrukcja - część 2</b></p>		<p>Eliminacja redundancji w projekcie. Udział w pracach projektowych i implementacji</p>
<p><b>15.04.19</b> 8- tydzień (3 dni) 9- tydzień</p>	<p>Kod programu w wersji SE i EE należy wysłać do dnia <b>23.04.19</b></p>	<p>Sprint planning meeting (45 min)</p>	<p>5) Tworzenie interfejsu graficznego użytkownika na platformie Java SE do programu zawierającego implementację warstwy biznesowej <b>Instrukcja 1 - opisująca przebieg prac podczas 1- 8 tygodnia</b></p>		<p>Eliminacja redundancji w projekcie. Udział w pracach projektowych i implementacji</p>

<p><b>25.04.19</b> 10-tydzień 11-tydzień 12-tydzień</p>	<p>Kod programu należy wystać do dnia <b>7.05.19</b> (wersja uzgodniona 25.04.19) oraz <b>14.05.19</b> (wersja końcowa uzgodniona 25.04.19)</p>	<p>Sprint planning meeting (45 min)</p>	<p>6) Przekształcenie aplikacji SE na aplikację działającą na platformie Java EE, zawierającej warstwę klienta typu desktopowego <b>Instrukcja 1 - opisująca przebieg prac podczas 1- 8 tygodnia</b> 7) Rozwijanie oprogramowania: projektowanie i implementacja oprogramowania w oparciu o wzorce projektowe – diagramy UML: rozwijanie diagramu klas, diagramy sekwencji i aktywności wybranych 2 (minimum) złożonych przypadków użycia reprezentujących projekt warstwy biznesowej projektu oraz implementacja projektu na platformie Java EE <u><b>Instrukcja 1 - opisująca przebieg prac podczas 1- 8 tygodnia</b></u> <u><b>Uwaga: działania wg instrukcji dodanej do tego etapu projektu</b></u></p>		<p>Eliminacja redundancji w projekcie. Udział w pracach projektowych i implementacji</p>
<p><b>16.05.19</b> 13- tydzień 14- tydzień</p>	<p>Kod programu wraz z testami należy wystać do dnia <b>21.05.19</b> (wersja robocza uzgodniona w dniu 16.05.19) i <b>28.05.19</b> (wersja do oceny)</p>	<p>Sprint planning meeting (45 min)</p>	<p>8) Sprawdzanie poprawności oprogramowania wg <u><b>Instrukcji do lab 10</b></u> i <u><b>Instrukcji do lab 11</b></u>. 9) Dodanie warstwy integracji z bazą danych opartą na technologii JPA (Java Persistence API) do programu oparta na automatycznym generowaniu kontrolerów typu Session Bean For Entity Class 10) <u><b>Uwaga: działania wg instrukcji dodanej do tego etapu projektu</b></u> 11) Uzpełnienie aplikacji desktopowej typu EE do wywołania operacji na bazie danych w technologii ORM.</p>		<p>Eliminacja redundancji w projekcie. Udział w pracach projektowych i implementacji</p>
<p><b>30.05.1913</b></p>		<p>Sprint planning meeting (45 min)</p>	<p>12) Zaliczenie projektu</p>		

## Stosowany proces wytwarzania oprogramowania - Scrum

Stosowany będzie proces wytwarzania oprogramowania wzorowany na metodyce Scrum.

Z metodyki SCRUM zaczerpnięte zostały następujące elementy:

- Daily Scrum of Scrums - krótkie spotkanie dotyczące aktualnego statusu projektu, w trakcie którego następuje synchronizacja prac wykonywanych przez poszczególne zespoły. W spotkaniu bierze udział prowadzący, Scrum Master i dokładnie jeden przedstawiciel każdej grupy projektowej, pozostali członkowie grup projektowych mogą brać udział w spotkaniu tylko jako obserwatorzy. W trakcie spotkania przedstawiciel zespołu powinien udzielić odpowiedzi na następujące trzy pytania:
  - o Co zespół zrobił od poprzedniego Daily Scrum of Scrums?
  - o Co zespół planuje zrobić do następnego Daily Scrum of Scrums?
  - o Co przeszkadza zespołowi w realizowaniu zaplanowanych zadań?
- Sprint - ograniczona czasowo do 4-5 tygodni iteracja (patrz Tab. *Przebieg realizacji projektu*) w trakcie której zespoły pracują nad przekształceniem przydzielonych przypadków użycia w nadającą się do przekazania klientowi funkcjonalność.
- User Stories – lista wymagań użytkownika („opis świata rzeczywistego”)
- Product Backlog – lista wymagań tworzonego oprogramowania (przynajmniej dla bieżącego sprintu)
- Sprint Planning – planowanie zadań do wykonania podczas bieżącego sprintu
- Sprint Backlog - lista zadań i przypadków użycia z przypisanymi punktami odzwierciedlającymi ich trudność, które mają zostać zrealizowane w trakcie sprintu (z oszacowaną pracochłonnością). Powstaje w systemie java.net w trakcie Sprint planning meeting.
- Sprint planning meeting - 45 minutowe spotkanie rozpoczynające każdy sprint. W trakcie spotkania definiuje się Sprint Backlog. W spotkaniu biorą udział Product Owner, Scrum Master oraz przedstawiciele każdej grupy projektowej.
- Sprint review meeting - 45 minutowe spotkanie podsumowujące sprint. W trakcie spotkania prezentowane oraz omawiane są funkcjonalności zrealizowane w trakcie kończącego się sprintu.
- Sprint retrospective meeting – 45 minutowe spotkanie podsumowujące projekt. W trakcie spotkania powinna odbyć się dyskusja dotycząca możliwości usprawnienia stosowanego procesu wytwarzania oprogramowania.

## Przebieg pracy

Przebieg pracy w projekcie wynika bezpośrednio z przedstawionego harmonogramu w tabeli 1: *Przebieg realizacji projektu*.

Cały projekt jest podzielony na 3 sprinty. Każdy sprint składa się z takich samych elementów. Zaczyna się spotkaniem Sprint planning meeting, w trakcie którego wybierana są zagadnienia do zrealizowania w trakcie sprintu, a kończy się spotkaniem Sprint review meeting, w trakcie którego prezentowane są uzyskane wyniki. W trakcie sprintu, raz w tygodniu odbywają się spotkania Daily Scrum of Scrums, na których raportowany jest aktualny status. Najważniejszym zadaniem w trakcie sprintu jest rozwiązywanie zaplanowanych na sprint zagadnień, czyli błędów, zadań i przypadków użycia.

Każdy wynik prezentowany podczas Daily Scrum of Scrums jest oceniane za pomocą przydzielanych punktów, podanych w tabeli 1, zarówno podgrupom, jak i kierownikowi (Scrum Master) Realizacja przypadku użycia może być stosunkowo zawiła i obejmuje zazwyczaj wiele różnych aktywności. W związku z tym zaleca się, aby kierownik zespołu wprowadził w systemie java.net zadania składające się na realizację danego przypadkiem użycia i przydzielał je członkom zespołu.

#### Role projektowe

Zespół Scrum zwykle składa się od 3 do 9 osób, reprezentujących różne umiejętności. Jeden projekt będzie realizowany przez połowę grupy studenckiej zapisanej na dany termin, czyli zespół projektowy będzie się składał z około 9 osób.

- Product Owner - rola pełniona przez prowadzącego zajęcia. Product Owner decyduje o priorytetach poszczególnych zadań, tym samym do niego należy rozstrzygający głos odnośnie zestawu zagadnień wybieranych do Sprint Backlog w trakcie Sprint planning meeting. Product Owner decyduje również o tym, czy dane zagadnienie zostało zrealizowane w wystarczającym stopniu i z wystarczającą jakością, aby mogło zostać uznane za zaliczone.
- Development Team: podział na 4-5-osobowe podgrupy o 2-3 osoby w podgrupie, odpowiedzialne za specyfikowanie wymagań, projektowanie aplikacji, programowanie i testowanie na poziomie testów jednostkowych
  - Kierownik - 1 osoba w podgrupie, odpowiedzialna za koordynowanie prac wewnątrz podgrupy (np. definiowanie niepuktowanych podzadań w java.net), osoba kontaktowa w komunikacji między podgrupami, równocześnie pełni rolę programisty / modelarza
  - Inspektor / tester – 1 osoba w podgrupie, odpowiedzialne za przeglądanie i weryfikowanie wymagań oraz modelu (przygotowują raport inspektora) oraz za testy jednostkowe.
- Administrator – 1 osoba w całym projekcie, odpowiedzialne za przygotowywanie i konserwowanie środowiska programistycznego ze szczególnym uwzględnieniem systemu ciągłej integracji oraz za raportowanie i przypisywanie błędów znalezionych przy pomocy systemu ciągłej integracji. Funkcja cykliczna, kadencja trwa 1 sprint, każdy kolejny administrator powinien pochodzić z innej 4-6-osobowej podgrupy.
- Scrum Master – ta sama osoba w projekcie, która pełni rolę Administrator. Każdy kolejny Scrum Master powinien pochodzić z innej 3-osobowej podgrupy. Scrum Master oprócz swojej funkcji powinien wykonywać zadania związane z implementacją realizowanego przez jego grupę przypadku użycia. Administruje systemem java.net, w szczególności ma uprawnienia do zmieniania osoby przypisanej do błędu.

#### Przygotowywane produkty:

- Faza modelowania
  - scenariusz PU, najlepiej w postaci diagramu aktywności, czas życia dokumentu - czas trwania całego projekt
  - specyfikacja testów akceptacyjne do PU, czas życia dokumentu - czas trwania całego projekt albo do momentu zautomatyzowania w postaci wykonywalnej w systemie ciągłej integracji
  - diagram sekwencji do PU o diagram klas do PU
  - \*diagram stanów (jeżeli jest potrzebny) do PU
  - raport inspektora (można generować z Java.net albo przygotowywać ręcznie na podstawie wspomnianego powyżej szablonu)

- Faza implementacji
  - kod (działająca aplikacja) – obowiązkowa separacja kodu warstwy biznesowej od warstwy prezentacji o zautomatyzowane testy jednostkowe (JUnit)
  - uruchomione manualnie lub zautomatyzowane w wybranym narzędziu testy jednostkowe
- Dokumenty wytwarzane poza fazami, czas życia dokumentu - czas trwania całego projektu o diagram PU o diagramy ilustrujące architekturę całego systemu, np.: diagram klas domenowych, diagram ilustrujący podział systemu na komponenty

Czas życia dokumentu wynosi 1 Sprint (kodu i zautomatyzowanych testów nie zaliczamy do dokumentów), chyba że napisano w opisie dokumentu inaczej. Dokumenty o czasie życia wynoszącym 1 Sprint nie muszą być aktualizowane po zakończeniu sprintu, w którym były wytwarzane. Pozostałe dokumenty muszą być aktualizowane aż do końca projektu. Odpowiedzialna za aktualizację dokumentu jest podgrupa, która wprowadziła zmianę, w efekcie której dokument musi zostać zaktualizowany.

#### Zalecane narzędzia

- UML 2.1 - Visual Paradigm 15.2 Community Edition (każdy diagram powinien powstawać w osobnym pliku)
- NetBeans 8.2, Java SE 8, Java EE 7