

<p align="center"><b>2019/20      WIZ, Informatyka</b></p> <p align="center"><b>Wspomaganie zarządzania projektami informatycznymi - laboratorium</b></p>			
<p>Rasz Arkadiusz Indeks 242493</p>	<p><b>Sprawozdanie do tematu nr 3:</b> Zarządzanie wymaganiami</p>		
<p>Termin zajęć: Środa 11:15-13:00</p>	<p>Data zajęć: 23.10.2019r.</p>	<p>Data oddania sprawozdania: 30.10.2019r.</p>	<p>Ocena:</p>

# 1. Wstęp teoretyczny

## 1.1. Zarządzanie wymaganiami

Zarządzanie wymaganiami to proces tworzenia cech systemu oraz kontrolowania ich zmian. Zajmuje się działaniami mającymi na celu przełożenie potrzeb lub oczekiwań użytkowników na fizyczną realizację produktu końcowego.

## 1.2. Etapy procesu zarządzania wymaganiami

Proces zarządzania wymaganiami możemy podzielić na 4 etapy:

- **Analiza otoczenia projektu**  
Na tym etapie odbywa się identyfikacja instytucji regulujących działanie podmiotu dla którego powstaje narzędzie, przetwarzanych danych oraz wszystkich interfejsów.
- **Pozyskiwanie wymagań**  
Na tym etapie określa się cele projektu, jego ogólny zarys, narzędzia oraz możliwości i ograniczenia
- **Analiza wymagań**  
Proces w którym uzgadniane są wymagania postawione przez klienta, ale też odkrywane są nowe wymagania
- **Zarządzanie zmianą wymagań**

## 1.3. Wymagania

Wymagania dzielą się na:

- Funkcjonalne  
Umożliwiają zidentyfikowanie i opisanie zachowania systemu oraz usługi które oferuje
- Niefunkcjonalne  
Określa, jaki powinien być system, co oferuje interfejs itp. Typowe wymagania niefunkcjonalne to wydajność, skalowalność, odporność na awarie, bezpieczeństwo.

Wymaganiami zarządza się w całym cyklu życia oprogramowania. Zmiana wymagań może nastąpić w każdej fazie wytwarzania oprogramowania. Wiele czynników może wpłynąć na taką zmianę. Oprogramowanie jest często wytwarzane iteracyjnie co nieuchronnie prowadzi do ich zmian.

Specyfikacja wymagań powinna być:

- Kompletna  
Zawiera pełen zbiór wymagań i każde z nich jest kompletnie opisane
- Jednoznaczna  
Zawiera wymagania tylko jednej semantycznej operacji
- Spójna  
Nie zawiera wymagań wzajemnie sprzecznych
- Weryfikowalna  
Zawiera procedury sprawdzające, czy wymagania zostały zrealizowane
- Modyfikowalna

Pozwala w łatwy sposób dokonać zmian w wymaganiach przy zachowaniu kompatybilności i spójności między nimi

- Sprawdzalna  
Pozwala na śledzenie wpływu wszelkich zmian na dane wymaganie poprzez rejestrację źródła każdego z wymagań

## 2. Wykorzystane oprogramowanie

### 2.1. Confluence

Confluence to komercyjna platforma firmy Atlassian służąca do zarządzania wiedzą, marketowana jako otwarta i współdzielona przestrzeń robocza. Główną funkcjonalnością jest tworzenie prostych stron służących do dokumentacji wymagań zorganizowanych w przestrzeni. Confluence jest uniwersalna i integruje się z innymi narzędziami Atlassiana, jak Jira i Trello.

### 2.2. Enterprise Architect

Enterprise Architect to narzędzie do modelowania i przygotowywania diagramów UML. Dokumentację wymagań wspiera za pomocą języka SysML. Zawiera szereg narzędzi do zapisu wymagań oraz ich powiązań z innymi elementami modelu. Wspomaga także śledowanie wymagań.

### 2.3. SpiraTeam

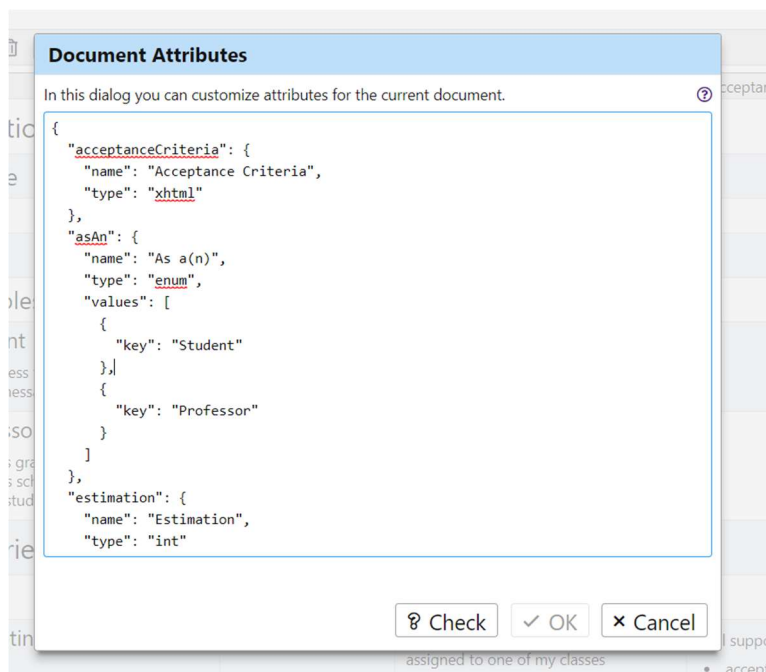
SpiraTeam to aplikacja webowa do zarządzania cyklem życia oprogramowania. Poza dokumentacją i zarządzaniem wymaganiami pomaga również w zarządzaniu testami, bugami oraz zadaniami.

### 2.4. ReqView

ReqView to narzędzie do zarządzania wymaganiami i ryzykiem. Główną funkcjonalnością jest tworzenie wymagań w postaci tabel, w których każdy wiersz ma swój unikalny identyfikator i może stanowić wymaganie lub sekcję dokumentu. Umożliwia to odnoszenie się do konkretnych wymagań lub ich zbiorów. Aplikacja jest prosta i elastyczna, wspiera wiele standardów, posiada wbudowane szablony. Wspierana jest śladowalność wymagań oraz śledzenie zmian.

## 3. Przebieg ćwiczenia




Naszym zadaniem było użycie narzędzia ReqView do krótkiego opisu wymagań i historyjek użytkownika dla elektronicznego systemu obsługi studentów. Ograniczyłem się do prostych przypadków użycia, jak wprowadzanie ocen, przeglądanie ich, akceptowanie i reklamowanie. Dokumenty utworzyłem z wbudowanych w program wzorców. W opcji zmiany atrybutów wprowadziłem użytkowników: Student oraz Profesor. Niestety, narzędzie nie udostępnia do takiej zmiany prawidłowego interfejsu użytkownika, zamiast tego zmiany wprowadza się w surowym formacie json.



Wprowadziłem proste wprowadzenie, opisy użytkowników oraz kilka historyjek użytkownika dla studenta oraz profesora:

Type	Description	As a(n)	I want	Acceptance Criteria
	Student has access to their classes, grades and can exchange messages with professors.			
Information	1.3.2 Professor Professor enters grades for classes he teaches, views his classes schedule and exchanges messages with students and other professors.			
Section	2 User Stories			
Epic	2.1 Grades			
User Story	2.1.1 Accepting a grade	Student	to accept a grade that a professor has assigned to one of my classes	Tool supports: • accepting an assigned grade
User Story	2.1.2 Making a complaint	Student	to make a complaint of a grade that i feel is not assigned correctly	Tool supports: • making a complaint on a grade • attaching a message to a grade
User Story	2.1.3 Entering grades	Professor	to enter grades for classes I teach	Tool supports: • browsing classes • browsing students signed up for a class • assigning grades for student in some class
Epic	2.2 Schedules			

Na podstawie historyjek użytkownika stworzyłem dokument z wymaganiami funkcjonalnymi projektu:

* ID	Description	Priority
FUN_REQ-1	<b>1 Functional Requirements</b>	
FUN_REQ-2 	<b>1.1 Entering grades</b> The tool supports entering grades to students for some class.	High
FUN_REQ-3 	<b>1.2 Accepting grades</b> The tool supports accepting grades that have been assigned.	High
FUN_REQ-4 	<b>1.3 Making a complaint</b> The tool provides a form for leaving a complaint with a message.	High
FUN_REQ-5	<b>1.4 Exporting schedule</b> The tool provides an export function for a whole schedule. File exported should have an .ical extension	Medium
FUN_REQ-6	<b>1.5 Browsing schedule</b> The tool provides a view for daily, weekly and monthly schedule with classes signed for.	Medium
FUN_REQ-7	<b>1.6 Browsing grades</b> The tool provides a view for all grades assigned to a student. Grades can be filtered by period.	Medium

Część z wymagań udało mi się połączyć z historyjkami użytkownika, co umożliwia śladowalność wymagań. Aplikacja ReqView jest jednak dosyć skomplikowana dla nowych użytkowników, przez co nie potrafiłem znaleźć opcji usunięcia utworzonego już połączenia. Utworzone dokumenty można wyeksportować do plików takich jak Excel, statyczny html oraz dokumenty programu Word.