

# **Wspomaganie zarządzania projektami informatycznymi**

Laboratorium

Maciej Kopiński

254578

## Spis treści

Z2.	Wykorzystanie oprogramowania do zarządzania wymaganiami.....	4
	Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	4
	Opis wykonanych zadań .....	4
	Podsumowanie .....	6
Z3.	Wykorzystanie oprogramowania do modelowania systemu z wykorzystaniem UML.....	6
	Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	6
	Opis wykonanych zadań .....	6
	Podsumowanie .....	7
Z4.	Wykorzystanie oprogramowania do modelowania systemu z wykorzystaniem UML.....	7
	Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	7
	Opis wykonanych zadań .....	8
	Podsumowanie .....	9
Z5.	Wykorzystanie oprogramowania do tworzenia i utrzymywania macierzy RACI.....	9
	Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	9
	Opis wykonanych zadań .....	10
	Podsumowanie .....	10
Z6.	Wykorzystanie oprogramowania do harmonogramowania projektu informatycznego.....	10
	Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	10
	Opis wykonanych zadań .....	11
	Podsumowanie .....	11
Z7.	Wykorzystanie oprogramowania do monitorowania realizacji projektu.....	12
	Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	12
	Opis wykonanych zadań .....	12
	Podsumowanie .....	13
Z8.	Wykorzystanie oprogramowania do przydzielania i rozliczania zadań .....	14
	Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	14
	Opis wykonanych zadań .....	14
	Podsumowanie .....	14
Z9.	Wykorzystanie oprogramowania do komunikacji w grupie.....	15
	Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	15
	Opis wykonanych zadań .....	15
	Podsumowanie .....	17
Z10.	Wykorzystanie oprogramowania do szacowania całkowitego kosztu nabycia i utrzymania oprogramowania .....	17
	Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	17

Opis wykonanych zadań .....	17
Podsumowanie .....	18
Z11. Wykorzystanie oprogramowania do zarządzania ryzykiem .....	18
Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	18
Opis wykonanych zadań .....	19
Podsumowanie .....	20
Z12. Oprogramowanie do zarządzania konfiguracją.....	20
Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	20
Opis wykonanych zadań .....	21
Podsumowanie .....	22
Z13. Wykorzystanie oprogramowania zintegrowanego w chmurze obliczeniowej.....	22
Nazwa, cel, funkcjonalność wykorzystanego oprogramowania.....	22
Opis wykonanych zadań .....	23
Podsumowanie .....	23

## Z2. Wykorzystanie oprogramowania do zarządzania wymaganiami.

Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

Na zajęciach dotyczących zarządzania wymaganiami omówione były potrzeby stosowania wymagań. Opisano przykładowe programy do definiowania wymagań:

- Rational DOORS Next
- Jira
- Micro Focus Dimensions

Największa uwaga była poświęcona oprogramowaniu Jira, czyli oprogramowanie do wspomagania zarządzania pracą zespołową od firmy Atlassian.

Pierwotnie służyła ona do śledzenia i naprawiania błędów, jednakże program rozrósł się do narzędzia, które pozwala na zarządzanie dowolnym projektem.

### Opis wykonanych zadań

1. Korzystając z Jiry i projektu Scrum utwórz wymagania do ulubionej gry. Wymagania muszą mieć określone priorytety (minimum 5 wymagań łącznie).

The screenshot displays the Jira interface. On the left, a Scrum board is visible with columns for 'Epik', 'LUT', and 'MAR'. A user story 'LAB2-8 Obsługa sklepu' is highlighted in the 'Epik' column. The right side of the image shows a detailed view of this user story. It includes a title 'Obsługa sklepu', a description field, and a list of child issues. The child issues are 'LAB2-11 Zakup przedmiotu' and 'LAB2-12 Sprzedaż przedmiotu w sklepie', both with a 'Lowest' priority. The interface also shows a progress bar for 'Child issues' at 0% completion.

2. Utwórz wymaganie śladowane wertykalnie na trzech poziomach.

LAB2-8
 
 1

## Obsługa sklepu

Do zrobienia ▾

Opis

Dodaj opis...

Child issues
 Kolejność wg: ▾ ... +

Gotowe 0%

☒ LAB2-11 Zakup pr... ▾ - DO ZROBIENIA ▾

☒ LAB2-12 Sprzedaż... ▾ - DO ZROBIENIA ▾

Połączone zgłoszenia +

relates to

LAB2-4 Obsługa int... ⚡ DO ZROBIENIA ▾

3. Na podstawie wymagania (np. epic): „Obsługa akcji użytkownika”, utwórz 4 bardziej szczegółowe wymagania.

Epik	LUT	MAR
<div> <div>LAB2-4 Obsługa interakcji użytkownika</div> <div> <input checked="" type="checkbox"/> LAB2-13 Obsługa kliknięć myszą DO ZROBIEN...                             <input checked="" type="checkbox"/> LAB2-15 Obsługa klawiatury DO ZROBIEN...                             <input checked="" type="checkbox"/> LAB2-14 Ruch kamery DO ZROBIEN...                             <input checked="" type="checkbox"/> LAB2-16 Obsługa ruchu myszy DO ZROBIEN...                         </div> <div> <div>LAB2-8 Obsługa sklepu</div> <div> <input checked="" type="checkbox"/> LAB2-11 Zakup przedmiotu DO ZROBIEN...                             <input checked="" type="checkbox"/> LAB2-12 Sprzedaż przedmiotu w sklepie DO ZROBIEN...                         </div> </div> <div>+ Utwórz Epik</div> </div>		

LAB2-4
 
 1

## Obsługa interakcji użytkownika

Do zrobienia ▾

Opis

Dodaj opis...

Child issues
 Kolejność wg: ▾ ... +

Gotowe 0%

☒ LAB2-13 Obsługa ... = - DO ZROBIENIA ▾

☒ LAB2-15 Obsługa ... = - DO ZROBIENIA ▾

☒ LAB2-14 Ruch ka... = - DO ZROBIENIA ▾

☒ LAB2-16 Obsługa ... = - DO ZROBIENIA ▾

## Podsumowanie

Poznanie programu Jira jest bardzo przydatne, ponieważ jest to oprogramowanie popularne i powszechnie stosowane w instytucjach. Posiada ono wiele zalet, jak np. dużą elastyczność, duże wsparcie, nowoczesny interfejs oraz zapewnia integrację z innymi narzędziami. Jira jest dostępna także w wersji darmowej jednak jest ona ograniczona i nie udostępnia wszystkich swoich funkcjonalności.

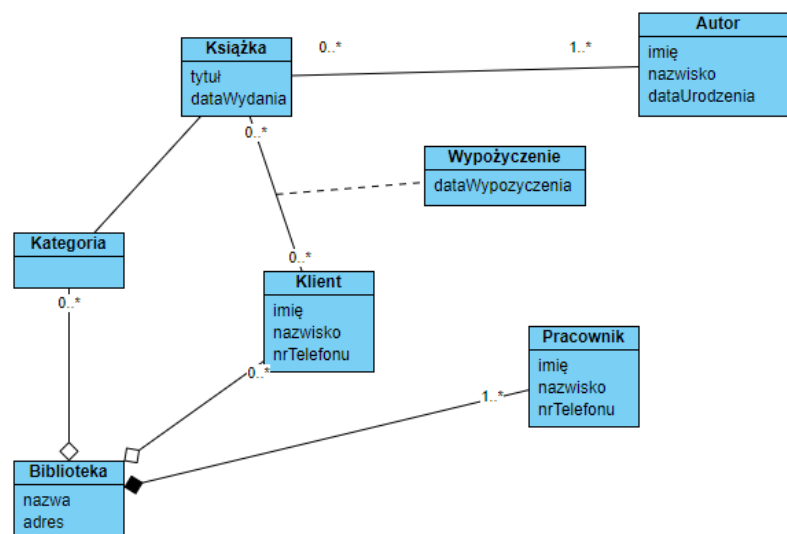
## Z3. Wykorzystanie oprogramowania do modelowania systemu z wykorzystaniem UML.

Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

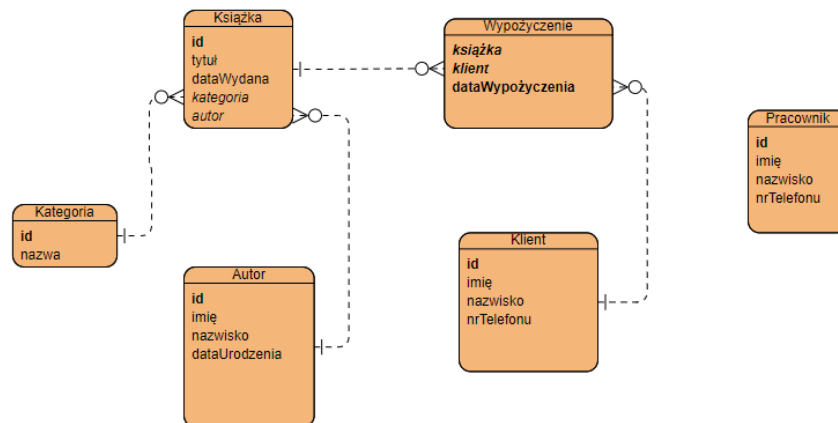
Na zajęciach o tematyce modelowania systemu z wykorzystaniem UML (Unified Modeling Language) przedstawione zostało oprogramowanie Visual Paradigm, a konkretnie jego darmowa wersja online. Oprogramowanie to służy do modelowania systemu dzięki wykorzystaniu wszelkiego rodzaju diagramów, jak np. diagram klas lub diagram encji w relacyjnej bazie danych.

### Opis wykonanych zadań

1. Utwórz diagram klas dla podanego schematu Biblioteki. Książka może posiadać wielu autorów, a autor może napisać wiele książek. Każda książka należy do jakiejś kategorii. Biblioteka posiada przynajmniej jednego pracownika i dowolną liczbę klientów oraz kategorii. Klienci i pracownicy posiadają imię, nazwisko i nr telefonu. Książka posiada tytuł i datę wydania. Biblioteka ma nazwę i adres. Autor ma imię, nazwisko i datę urodzenia. Klient może wypożyczyć dowolną liczbę książek, ale należy zapisać datę wypożyczenia.



2. Diagram klas z zadania pierwszego należy przerobić na diagram encji dla bazy danych.



## Podsumowanie

Oprogramowanie do modelowania systemu może okazać się przydatne, jeśli chcemy wstępnie zwizualizować dany proces lub reprezentację danych czy też samego systemu. Możemy go także użyć do zdefiniowania konkretnych reguł, których będziemy się następnie trzymać przy implementacji systemu.

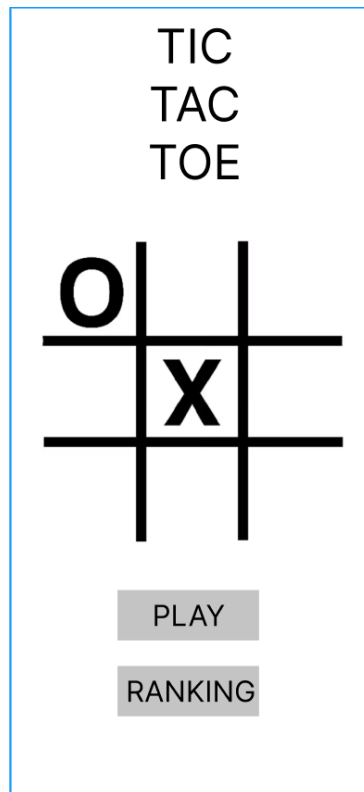
## Z4. Wykorzystanie oprogramowania do modelowania systemu z wykorzystaniem UML.

Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

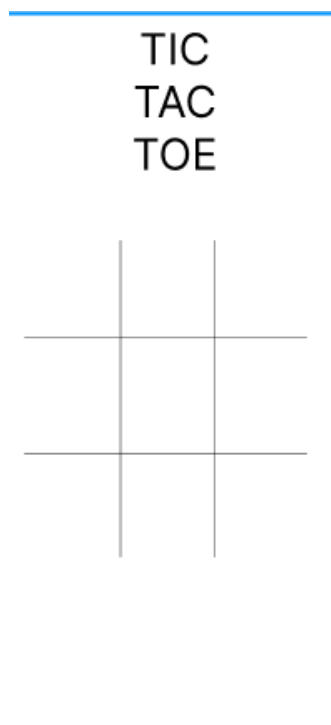
Na zajęciach o tematyce modelowania interfejsów zostały omówione różnice pomiędzy różnymi rodzajami samych interfejsów jak i różnice pomiędzy mockupami, wireframe'ami oraz prototypami. Zostały także omówione przykładowe programy do modelowania interfejsów. Główna uwaga została jednak poświęcona oprogramowaniu Figma.

## Opis wykonanych zadań

1. Przygotuj pierwszą stronę gry kółko i krzyżyk (tytuł, przyciski – gra, ranking).

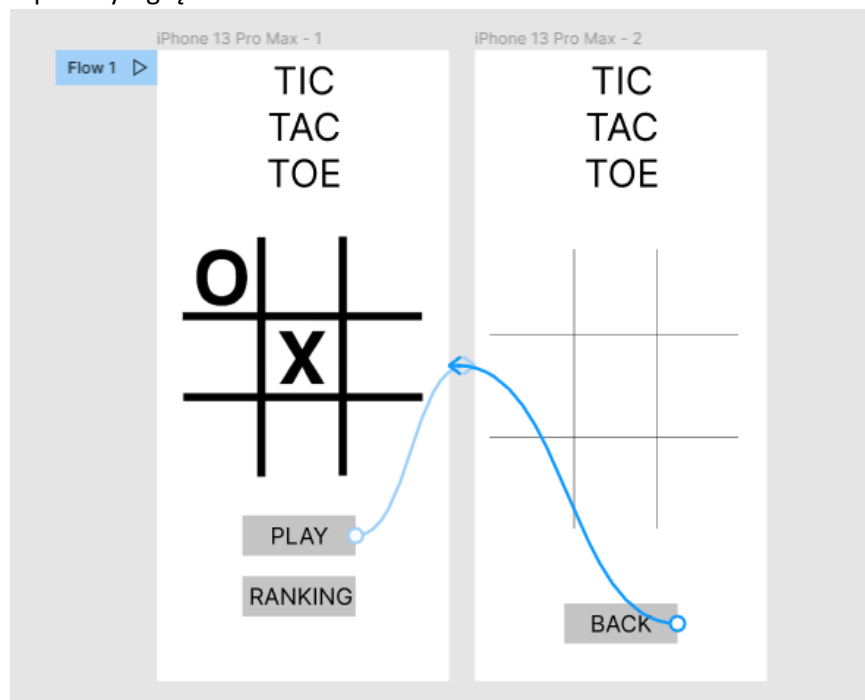


2. Przygotuj stronę z pustą planszą do gry.





3. Dodaj przejście między ekranami, po nadczeniu przycisku na stronie głównej i przycisk powrotu na planszy z grą.



## Podsumowanie

Oprogramowanie do modelowania interfejsu użytkownika jest nam potrzebne do wstępnego zaplanowania tworzonej aplikacji bądź w celu przedstawienia konceptu potencjalnemu klientowi.

## Z5. Wykorzystanie oprogramowania do tworzenia i utrzymywania macierzy RACI.

### Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

Programami służącymi do tworzenia i utrzymywania macierzy RACI przedstawionymi na zajęciach były Asana oraz Jira. Istnieje jednak możliwość tworzenia takiej macierzy w programach takich jak: Excel, Visual Paradigm Online, Creately oraz Lucidchart. Wszystkie z nich pozwalają na utworzenie macierzy, która służy do określania ról w zespołach projektowych oraz odpowiedzialności za poszczególne zadania.

## Opis wykonanych zadań

1. Przygotuj macierz RACI dla procesu budowania bloku mieszkaniowego dla osób:
  - a. Klient (sponsor)
  - b. Architekt
  - c. Kierownik budowy
  - d. Budowniczy
2. Oraz zadań:
  - a. Zaprojektowanie budynku
  - b. Określenie wymagań projektowych
  - c. Postawienie fundamentów
  - d. Wykończenie wnętrza

RACI	Klient	Architekt	Kierownik	Budowniczy
Określenie wymagań projektowych	R	C	A + C	I
Zaprojektowanie budynku	C	R	A	I
Postawienie fundamentów	C	I	A	R
Wykończenie wnętrza	C	I	A	R

## Podsumowanie

Oprogramowanie do tworzenia i utrzymywania macierzy RACI pomaga nam jasno określić rolę osób w zespole, ułatwia współpracę pracowników i przyspiesza proces komunikacji między nimi lecz jest także mało intuicyjna oraz ciężko utrzymywać jej aktualną wersję.

## Z6. Wykorzystanie oprogramowania do harmonogramowania projektu informatycznego.

### Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

Harmonogramowanie projektu informatycznego pozwala nam na identyfikację zadań oraz zależności pomiędzy nimi., oszacowanie czasu trwania oraz stworzenie diagramu sieciowego.

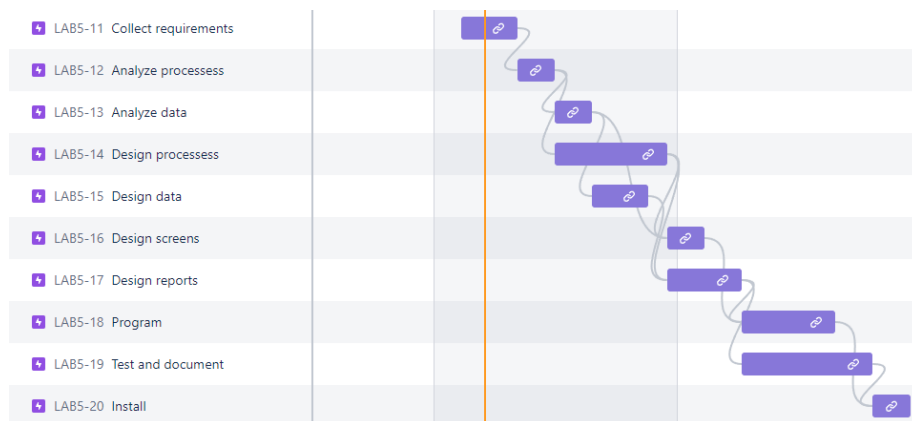
Oprogramowaniem do harmonogramowania projektu przedstawionym na zajęciach była Jira, która pomaga na utworzenie wykresu Gantta. Innymi programami, które były wspomniane na zajęciach były: Aha! oraz Microsoft Project.

## Opis wykonanych zadań

1. Na podstawie tabelki utwórz wykres Gantta w Jira(roadmaps) lub innym programie.



2. Utwórz wykres Gantta na podstawie tabelki, dla 3 aktywności zdefiniuj po 2 podzadania, czas ich wykonania można podzielić dowolnie.



## Podsumowanie

Narzędzia do harmonogramowania projektu informatycznego pozwalają nam ustalić sekwencję wykonywania zadań, ustalić zależności pomiędzy nimi oraz zadeklarować czas rozpoczęcia i zakończenia prac, a także zdefiniować bufor czasowe w zadaniach i projekcie.

## 27. Wykorzystanie oprogramowania do monitorowania realizacji projektu

Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

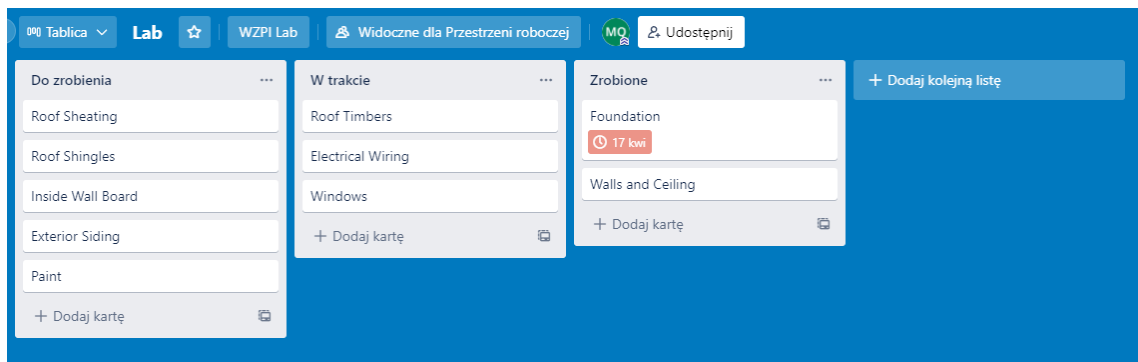
Celem oprogramowania do monitorowania realizacji projektu jest:

- Oszacowanie aktualnego ryzyka,
- Koordynacja i synchronizacja zadań częściowych,
- Kontrola kosztów, zakresu projektu oraz czasu,
- Minimalizowanie odczuć subiektywnych w ocenie stanu projektu
- Przewidywanie konfliktów i zakłóceń przebiegu projektu


Oprogramowaniem przedstawionym na zajęciach było Trello – bezpłatna dla indywidualnych użytkowników aplikacja.


Opis wykonanych zadań

1. Wykorzystując Trello, stwórz na podstawie poniższej tabeli tablicę z trzema listami: Do zrobienia, W trakcie, Zrobione. Jako datę początkową projektu przyjmij 17.04.2022. Zadania przyporządkuj do odpowiednich list. Dodaj daty rozpoczęcia i zakończenia kolejnych zadań oraz oznacz zakończone.



2. Do karty Electrical Wiring dodaj listę zadań, składającą się z trzech zadań: Purchase of wires and necessary parts, Wiring installation, Wiring safety tests. Oznacz dwa pierwsze zadania z listy jako wykonane. Do wybranej przez siebie karty dodaj co najmniej dwuelementową listę zadań.

 **Electrical Wiring**  
na liście [W trakcie](#)

 Opis

Dodaj bardziej szczegółowy opis...

☒ **Lista zadań**

Ukryj zaznaczone elementy

Usuń


67%

☒ Purchase of wires and necessary parts


☒ Wiring installation


☐ Wiring safety tests

Dodaj element...

 **Aktywność**

Pokaż Szczegóły

 **Paint**  
na liście [Do zrobienia](#)

 Opis

Dodaj bardziej szczegółowy opis...

☒ **Lista zadań**


Usuń

0%

☐ Purchase of paint

☐ Painting

Dodaj element...

 **Aktywność**

Pokaż Szczegóły

## Podsumowanie

Narzędzia do monitorowania realizacji projektu pozwalają nam wygodnie kontrolować pracę i umożliwiają informowanie wszystkich pracowników o postępie prac, co prowadzi do sprawnej realizacji.

## Z8. Wykorzystanie oprogramowania do przydzielania i rozliczania zadań

Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

Stosując system zarządzania zadaniami, można monitorować postępy, przydzielać zadania różnym członkom zespołu, ustalać terminy i zapewniać, że praca jest wykonywana w sposób wydajny i skoordynowany. Na zajęciach wspomniane były trzy oprogramowania do przydzielania i rozliczania zadań: Jira Software, GitHub oraz asana, natomiast dokładnie zostało zaprezentowane oprogramowanie GitHub.

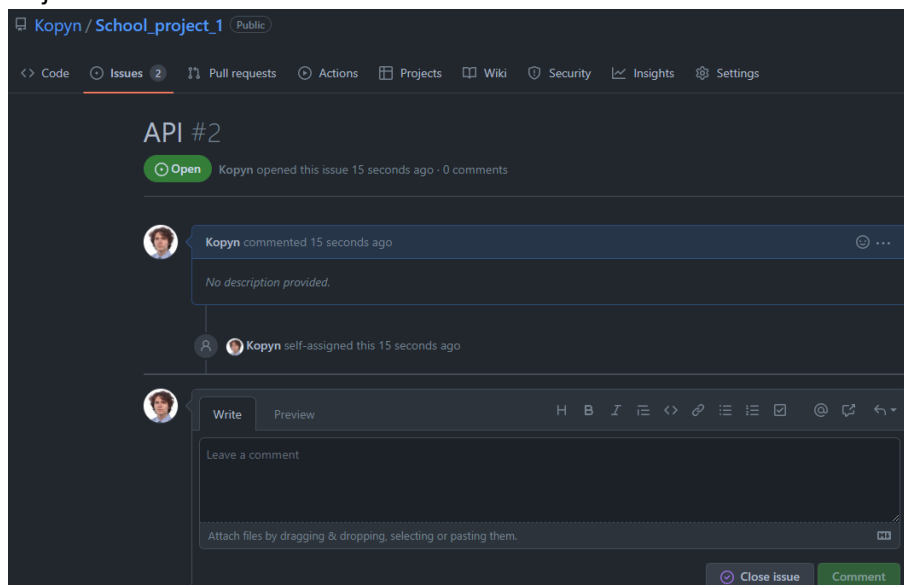
Opis wykonanych zadań

1. Stwórz na githubie projekt, dodaj do niego zakładkę z priorytetem, zdefiniuj własne statusy i priorytety.
2. Dodaj dwa dowolne zadania, wybierz osoby realizujące je i ich priorytet.



	Title	Assignees	Status	Priority	
1	API	Kopyn	In Progress	Medium	
2	Frontend	Kopyn	Done	Low	

3. Przekształć jedno z zadań w Issue.



Podsumowanie

Oprogramowanie do przydzielania i rozliczania zadań jest niezbędne do poznania, ponieważ jest powszechnie stosowane i niezbędne do zarządzania projektami.

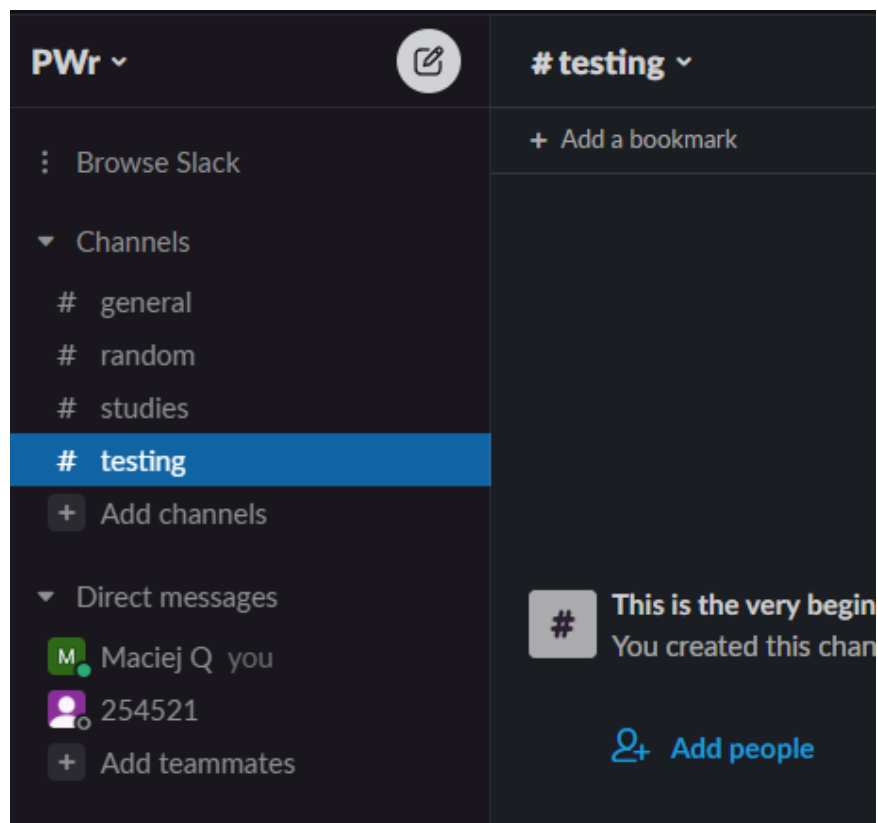
## Z9. Wykorzystanie oprogramowania do komunikacji w grupie

Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

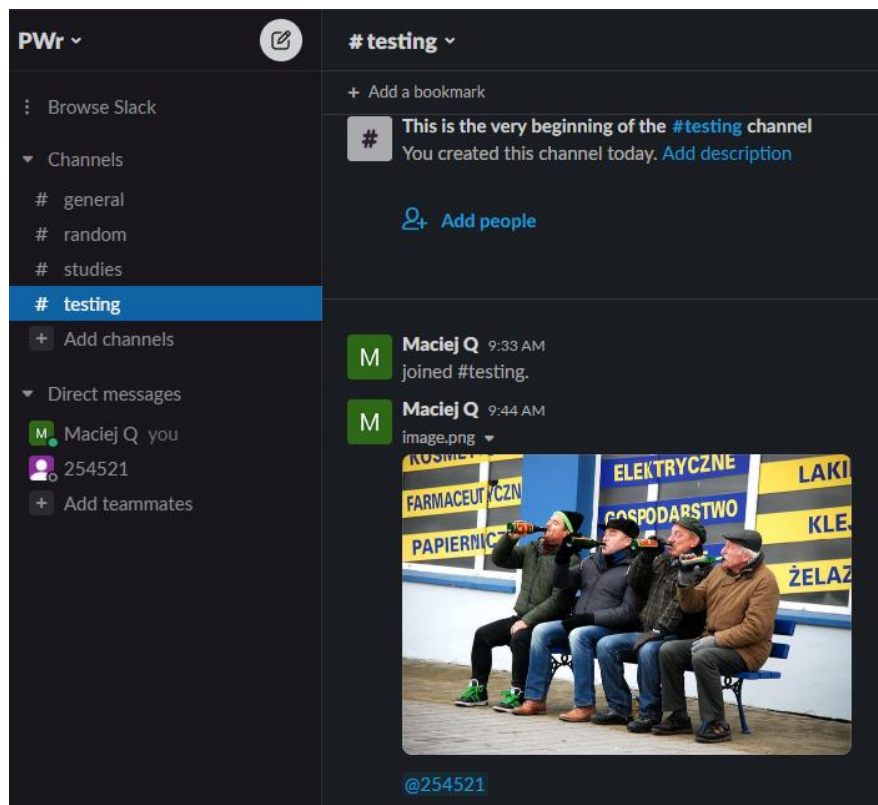
Oprogramowanie do komunikacji w grupie jest niezbędne do efektywnej i skutecznej pracy zespołowej. Na zajęciach, przedstawiono przykładowe programy takie jak: Microsoft Teams, Signal, Slack oraz Discord. Szczegółowo przedstawiony został trzeci z nich – Slack.

Opis wykonanych zadań

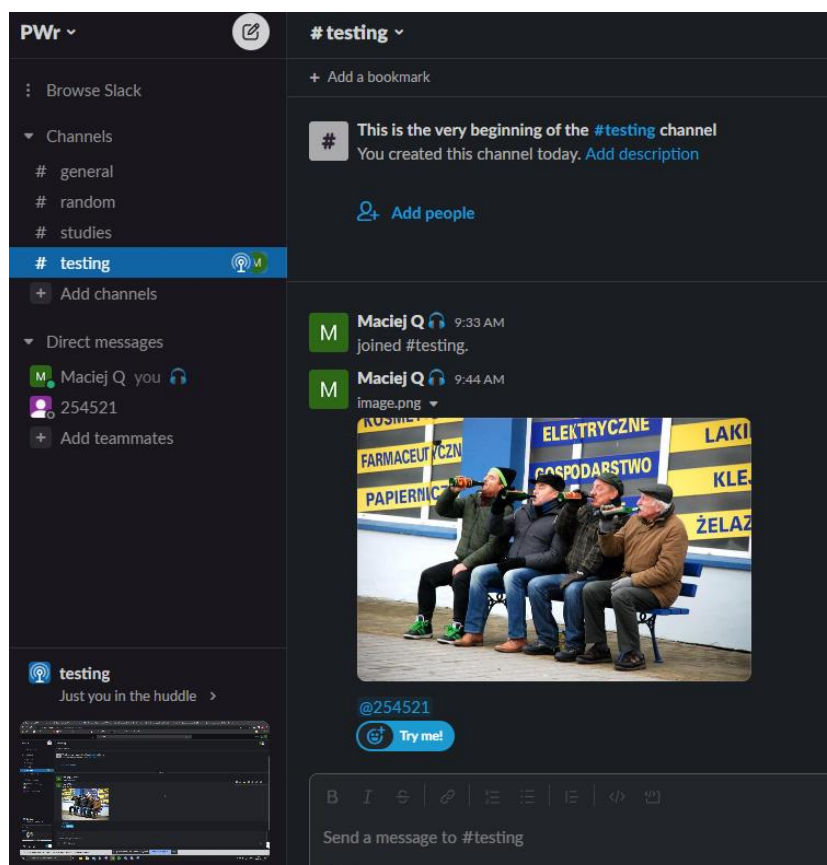
1. Stwórz serwer dla projektu oraz dodaj do nowych członków. Dodaj nowy kanał tekstowy o dowolnej nazwie (np. testing).



2. Wyślij plik poprzez kanał tekstowy oraz wspomnij na czacie o innym użytkowniku.



3. Dołącz do rozmowy i udostępnij swój ekran.





## Podsumowanie

Od dwóch lat narzędzia do komunikacji stały się niezwykle popularne ze względu na spopularyzowaniu pracy w formie zdalnej. W przypadku takiego systemu pracy niezbędne jest narzędzie pozwalające na kontakt członków zespołu.

## Z10. Wykorzystanie oprogramowania do szacowania całkowitego kosztu nabycia i utrzymania oprogramowania

### Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

Oprogramowanie do szacowania całkowitego kosztu nabycia i utrzymania oprogramowania pomagają wyliczyć różnego rodzaju koszty projektów: koszty osobowe, koszty sprzętu i oprogramowania oraz koszty realizacji przedsięwzięcia. Przykładowymi narzędziami służącymi do właśnie takiego szacowania są kalkulatory COCOMO.

### Opis wykonanych zadań

1. Za pomocą narzędzia do algorytmu COCOMO dokonaj wyceny nabycia i utrzymania projektu informatycznego o rozmiarze 3000 linii kodu:
  - Załóż, że 1000 linii kodu będzie reużywanych, z czego 40% wymaga integracji.
  - Zostaw współczynniki ustawione na "Nominal" lub ustaw według własnego uznania.
  - W ramach dodatkowej wyceny utrzymania (Maintenance - On), załóż, że rocznie modyfikowanych będzie 500 linii kodu, a czas wsparcia będzie wynosił 3 lata.
  - Ustaw koszt osobomiesiaca na tyle, ile sam/a byś chciał/a zarabiać miesięcznie

#### Results

##### Software Development (Elaboration and Construction)

Effort = 6.7 Person-months  
Schedule = 6.7 Months  
Cost = \$33588

##### Staffing Profile

Your project is too small to display a staffing profile due to truncation.

Total Equivalent Size = 2120 SLOC  
Effort Adjustment Factor (EAF) = 1.00

##### Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.4	0.8	0.5	\$2015
Elaboration	1.6	2.5	0.6	\$8061
Construction	5.1	4.2	1.2	\$25527
Transition	0.8	0.8	1.0	\$4031

##### Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.1	0.2	0.5	0.1
Environment/CM	0.0	0.1	0.3	0.0
Requirements	0.2	0.3	0.4	0.0
Design	0.1	0.6	0.8	0.0
Implementation	0.0	0.2	1.7	0.2
Assessment	0.0	0.2	1.2	0.2
Deployment	0.0	0.0	0.2	0.2

##### Maintenance

Annual Maintenance Effort = 1.4 Person-Months  
Annual Maintenance Cost = \$6859  
Total Maintenance Cost = \$20577

2. Zaproponuj temat aplikacji i ogólny zakres jej funkcjonalności (platformy iOS, Android, iOS+Android, Web). Następnie użyj kalkulatora na stronie Fulcrum do oszacowania liczby roboczogodzin (opcjonalnie możesz podać także adres email, aby otrzymać raport z wyceną - skalibrowaną na rynek amerykański).

## YOUR PRICE ESTIMATE

Great! Your price estimate is calculated. Enter your email. We will send you the detailed report shortly.

Platform: [Web](#)

Title: \_\_\_\_\_ Time: \_\_\_\_\_

### Admin Features

Ready-made admin 24h

### Content Generation

Gallery - Photo / Video 40h

Content creation 80h

### Content Representation

Search 40h

Maps view 160h

### E-Commerce

Total hours

**942 - 1275**

Estimate hours may vary, this price is an approximate.

Write your email to receive a commercial offer by mail.

Email

Your Email

[LEARN THE COSTS](#)

## Podsumowanie

Narzędzia pozwalające na oszacowanie kosztów projektu mogą okazać się bardzo przydatne i zapobiegają problemom związanym z niedoszacowaniem lub zawyżeniem kosztów, które mogą odbić się na realizacji projektu.

## Z11. Wykorzystanie oprogramowania do zarządzania ryzykiem

### Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

Oprogramowanie do zarządzania ryzykiem pozwala analizować i planować działania, które powinny zostać podjęte w przypadku wystąpienia problemu, jak np. awaria serwera. Przykładowymi programami służącymi do tego celu były przedstawione na zajęciach: Task, Resolver lub Jira. Największą uwagę poświęciliśmy jednak oprogramowaniu Jira ze względu na jej popularność.

## Opis wykonanych zadań

0. Dodaj plugin.
1. Dodaj w Backlogu 3 przykładowe ryzyka, np. awaria serwera, meteoryt spadł na siedzibę firmy.

▼ Backlog (3 zgłoszenia) 0 1 2 Utwórz sprint

<input checked="" type="checkbox"/>	LAB5-25 Server crash	DO ZROBIENIA ▼	
<input checked="" type="checkbox"/>	LAB5-26 Meteorite hit	DO ZROBIENIA ▼	
<input checked="" type="checkbox"/>	LAB5-27 Epidemy	DO ZROBIENIA ▼	

2. W ustawieniach Risk Register dodaj po jednym własnym poziomie prawdopodobieństwa i poziomu wpływu i odpowiednio zmodyfikuj wzór macierzy ryzyka.

**IMPACT →**

		Unnoticeable	Negligible	Low	Medium	High	Severe
<b>←PROBABILITY</b>	Very Unlikely	Low ▼	Low ▼	Low ▼	Low ▼	Medium ▼	Medium ▼
	UnLikely	Low ▼	Low ▼	Low ▼	Medium ▼	Medium ▼	High ▼
	Likely	Low ▼	Low ▼	Medium ▼	Medium ▼	High ▼	High ▼
	Very Likely	Low ▼	Medium ▼	Medium ▼	High ▼	High ▼	Extreme ▼
	Almost Certain	Low ▼	Medium ▼	High ▼	High ▼	Extreme ▼	Extreme ▼
	Certain	Medium ▼	Medium ▼	High ▼	Extreme ▼	Extreme ▼	Extreme ▼

3. Ustaw prawdopodobieństwa i wpływ dla zdefiniowanych wcześniej ryzyk oraz zobacz jak wygląda wtedy macierz ryzyka.

▼ Backlog (3 zgłoszenia) 0 1 2 Utwórz sprint

<input checked="" type="checkbox"/>	LAB5-25 Server crash	DO ZROBIENIA ▼	
<input checked="" type="checkbox"/>	LAB5-26 Meteorite hit	DO ZROBIENIA ▼	
<input checked="" type="checkbox"/>	LAB5-27 Epidemy	DO ZROBIENIA ▼	

+ Utwórz zgłoszenie

**Risk Register**

	Probability	Impact	Risk
Initial	Certain ▼	Unnoticeable ▼	Medium
Residual	Very Unlikely ▼	Unnoticeable ▼	Low

## Podsumowanie

Oprogramowanie do zarządzania ryzykiem może okazać się kluczowe w przypadku wystąpienia problemu. Jeśli wystąpienie takiego problemu było przez nas przewidziane na etapie planowania ryzyka, będziemy w stanie szybko reagować co będzie skutkowało mniejszymi stratami dla przedsiębiorstwa.

## Z12. Oprogramowanie do zarządzania konfiguracją

### Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

Na zajęciach zostały przedstawione trzy oprogramowania służące do zarządzania konfiguracją:

- GitHub Actions
- Ansible
- Puppet

Oprogramowanie to, pozwala nam śledzić i kontrolować wszelkie zmiany w oprogramowaniu.

## Opis wykonanych zadań

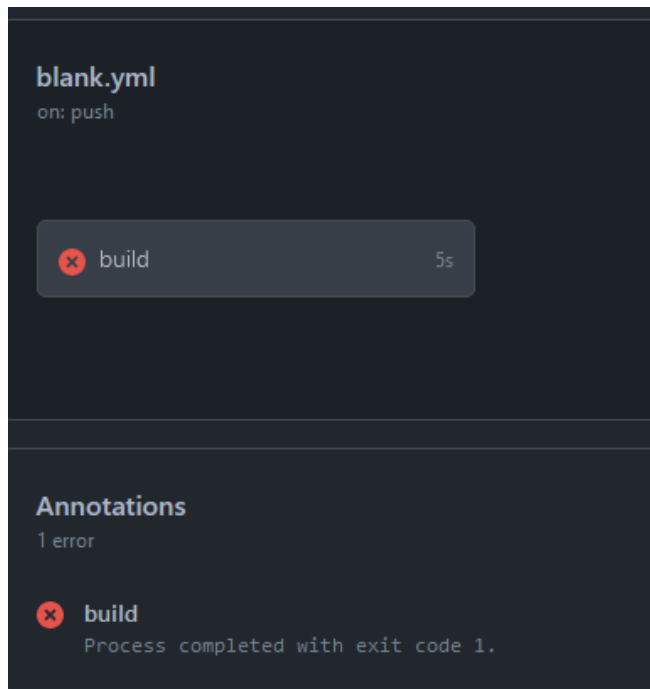
### 1. Utworzenie GitHub Action

- Utworzyć repozytorium na GitHubie (może być wymagane publiczne w razie braku konta Pro)
- Dodać nowe Action na podstawie „Simple workflow”
- W pliku .yml dodać nowe zadanie wypisujące do konsoli wynik polecenia „ls”
- Dodać przykładowy skrypt Python do repozytorium
- Dodać nowe zadanie wykonujące dodany skrypt

```
1 # This is a basic workflow to help you get started with Actions
2
3 name: CI
4
5 # Controls when the workflow will run
6 on:
7   # Triggers the workflow on push or pull request events but only for the "master" branch
8   push:
9     branches: [ "master" ]
10  pull_request:
11    branches: [ "master" ]
12
13 # Allows you to run this workflow manually from the Actions tab
14 workflow_dispatch:
15
16 # A workflow run is made up of one or more jobs that can run sequentially or in parallel
17 jobs:
18   # This workflow contains a single job called "build"
19   build:
20     # The type of runner that the job will run on
21     runs-on: ubuntu-latest
22
23     # Steps represent a sequence of tasks that will be executed as part of the job
24     steps:
25       # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
26       - uses: actions/checkout@v3
27
28       # Runs a single command using the runners shell
29       - name: Run a one-line script
30         run: echo Hello, world!
31
32       # Runs a set of commands using the runners shell
33       - name: Run a multi-line script
34         run: |
35           echo Add other actions to build,
36           echo test, and deploy your project.
37
38       # Runs a list command
39       - name: Run a list command
40         run: ls
41
42       # Runs python script
43       - name: Run python script
44         run: python test.py
```

## 2. Zastosowanie Actions do wykrywania błędów

- Wprowadzić błąd do skryptu Python (np. literówkę)
- Zaobserwować czy wykonanie Action zakończy się niepowodzeniem
- W kolejnej zmianie naprawić skrypt i potwierdzić, że Action poprawnie się wykonuje



## Podsumowanie

Oprogramowanie służące do zarządzania konfiguracją, śledzące zmiany w projekcie jest niezwykle przydatne, ponieważ pozwala przykładowo na regularną walidację funkcjonalności wykonywanego projektu. Pozwala to na wygodne i szybkie rozwiązywanie występujących problemów.

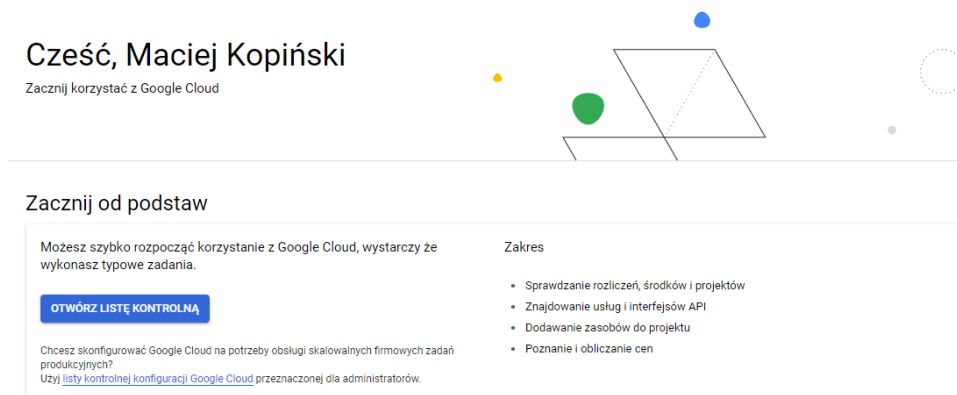
## Z13. Wykorzystanie oprogramowania zintegrowanego w chmurze obliczeniowej

### Nazwa, cel, funkcjonalność wykorzystanego oprogramowania

Oprogramowanie działające w chmurze może mieć dowolną funkcjonalność, natomiast sama chmura pozwala korzystać z oprogramowania bądź też zasobów obliczeniowych czy całych systemów, znajdujących się na zdalnych serwerach, uniezależniając tym samym użytkownika od własnego oprogramowania czy zasobów. Oprogramowaniem oferującym różne funkcje (oprogramowanie lub zasoby obliczeniowe) przedstawionym na zajęciach było Google Cloud.

## Opis wykonanych zadań

1. Przejdź proces logowania do Google Cloud.



2. Włącz moduł Compute Engine.
3. Stwórz maszynę wirtualną w tym module. Powinna być to maszyna Ubuntu LTS 18.04.

Filtruj Wpisz nazwę lub wartość właściwości

<input type="checkbox"/>	Stan	Nazwa ↑	Strefa	Zalecenia	Używany przez	Wewnętrzny adres IP	Zewnętrzny adres IP	Połącz
<input type="checkbox"/>	✓	ubuntu	us-west4-b			10.182.0.2 (nic0)	34.125.157.191 (nic0)	SSH ▾ ⋮

Powiązane działania

4. Napisz program w pythonie.

```
maciekkopinski69@ubuntu:~$ touch main.py
maciekkopinski69@ubuntu:~$ nano main.py
maciekkopinski69@ubuntu:~$ python main.py

Command 'python' not found, but can be installed with:

apt install python3
apt install python
apt install python-minimal

Ask your administrator to install one of them.

You also have python3 installed, you can run 'python3' instead.

maciekkopinski69@ubuntu:~$ python3 main.py
My python app
maciekkopinski69@ubuntu:~$
```

## Podsumowanie

Oprogramowanie w chmurze niesie za sobą wiele korzyści, jak np. obniżenie kosztów związanych z zakupem czy zarządzaniem własną infrastrukturą, czy poprawienie dynamiki w firmie poprzez dostęp do oprogramowania, działającego w chmurze, bez konieczności instalacji na każdej maszynie, która będzie z takiego oprogramowania korzystać.