

## **Inżynieria oprogramowania**

### **Część 2: Harmonogram realizacji projektu informatycznego**

1. HARMONOGRAM.....	3
1.1. Proces tworzenia harmonogramu.....	3
1.2. Pojęcia i technika tworzenia harmonogramu .....	4
1.3. Opis powiązań między zadaniami .....	4
2. Wykres Gantt'a .....	5
2.1. Etapy .....	5
2.2. GanttProject - Funkcje menu .....	6
2.3. Import zadań.....	6
przykład nr 1 do realizacji .....	6
2.4. Dodawanie ręczne zadania .....	7
2.5. Właściwości zadania.....	7
2.6. Modyfikacja parametrów czasowych zadania .....	7
Przykład 2 do realizacji.....	7
2.7. Role w projekcie .....	8
2.8. Osoby i role w projekcie.....	9
Dodawanie osoby do projektu.....	9
Niestandardowe role .....	10
Przykład nr 2 do realizacji .....	10
Przykład 3 do realizacji .....	10

## 1. HARMONOGRAM

Harmonogram to określony w czasie porządek realizacji zadań w projekcie. Głównymi składowymi harmonogramu są zadania, zależności między nimi, czas trwania oraz alokacja zasobów do poszczególnych zadań.

Jednym z trzech podstawowych parametrów, który definiuje i jednocześnie ogranicza projekt jest czas, któremu w planowaniu projektu i jego monitorowaniu poświęca się szczególną uwagę. Najczęściej mamy do czynienia z sytuacją dysponowania określonymi (najczęściej ograniczonymi) zasobami ludzkimi lub mamy narzucony czas na wykonanie projektu. Charakter projektu i technologia jego realizacji wpływa na związki oraz kolejność realizacji zadań.

### 1.1. PROCES TWORZENIA HARMONOGRAMU

Harmonogram jest to określony w czasie porządek realizacji zadań w projekcie. Głównymi składowymi harmonogramu są zadania, zależności między nimi, czas trwania oraz alokacja zasobów do poszczególnych zadań. Czas trwania realizacji zadania obliczamy według następującego wzoru:

$$\text{czas trwania zadania} = \text{wymagana praca} / \text{nakład pracy zasobu}$$

gdzie:

- czas trwania zadania jest rzeczywistą wielkością czasu, który jest planowany na wykonanie zadania (np. 5 dni),
- wymagana praca jest wielkością mierzoną w jednostkach czasochłonności niezbędnej do wykonania zadania (np. 4 osobogodziny),
- nakład pracy zasobu jest wielkością wyrażoną w jednostkach pracochłonności z uwzględnieniem tylko tego czasu, w którym zasób pracuje na rzecz danego zadania – jest alokowany.

Przykład

- Trzej programiści pracują nad zadaniem przez dwa dni przy nakładzie pracy 8 godzin dziennie, praca każdego zasobu wynosi 16 godzin: (2 dni · 8 godzin).
- Całkowity nakład pracy zasobów wynosi 24 godziny dziennie: (3 programistów · 8 godzin).
- Całkowita praca w zadaniu wynosi 48 godzin: (2 dni · 8 godzin · 3 programistów).
- Czas trwania zadania wynosi 2 dni: 48 godzin / (3 programistów · 8 godzin).

Zrozumienie powyższego wzoru jest ważne do oszacowania, w jaki sposób zmiany dokonywane w zadaniach wpływają na harmonogram projektu.

Prace nad harmonogramem związane są z wykonaniem następujących kroków:

- tworzenie hierarchicznej struktury zadań – WBS,
- specyfikacja zadań na podstawie WBS,
- szeregowanie zadań,
- tworzenie powiązań i zależności między zadaniami,
- określenie wymaganych zasobów,
- szacowanie pracochłonności,
- określenie czasu trwania zadania,
- stworzenie wstępnego harmonogramu projektu,
- stworzenie harmonogramu projektu,
- weryfikacja i korekta harmonogramu.

## 1.2. POJĘCIA I TECHNIKA TWORZENIA HARMONOGRAMU

Stosuje się najczęściej dwa podejścia co do przyjmowanego czasu trwania zadania:

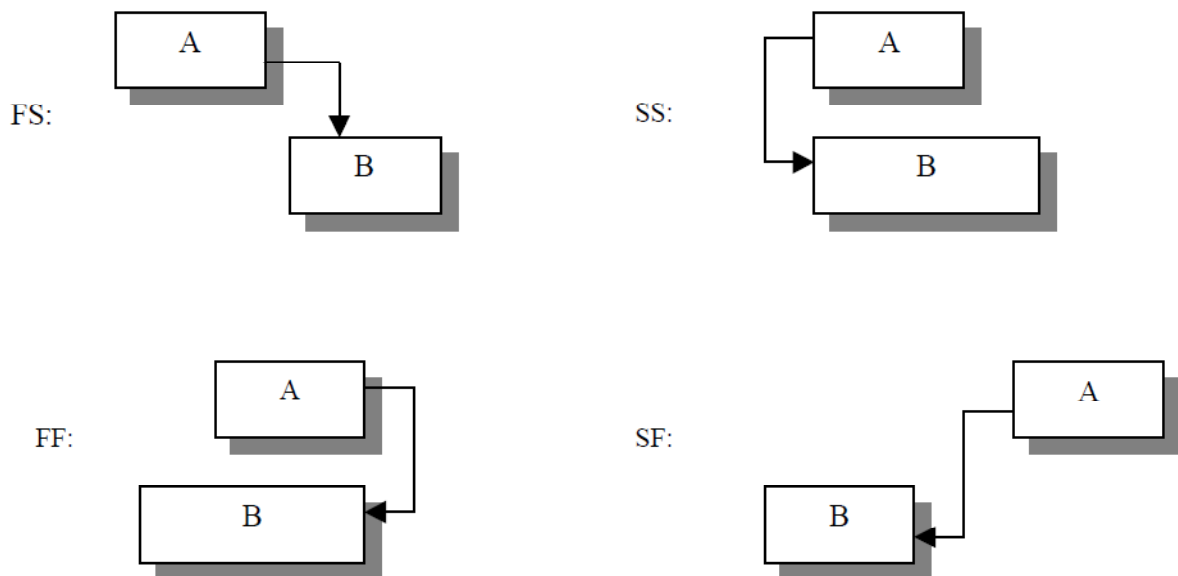
- zależne od posiadanych zasobów (ang. resource-driven scheduling),
- o ustalonym czasie minimalnym, w którym zadanie może być wykonane.

Wspomniana technologia realizacji i charakter projektu bezpośrednio wpływa na zależności między zadaniami, które wyspecyfikowano, aby zrealizować projekt.

## 1.3. OPIS POWIĄZAŃ MIĘDZY ZADANIAMI

Graficzne rozmieszczenie zadań na osi czasu oraz ich wzajemne powiązania przedstawia się na ogół w sposób, jak na Rysunku 1.

- Koniec – Start (ang. Finish-to-Start FS) – zadanie B nie może rozpocząć się przed ukończeniem zadania A,
- Start – Start (ang. Start-to-Start SS) – zadanie B nie może rozpocząć się przed rozpoczęciem zadania A,
- Koniec – Koniec (ang. Finish-to-Finish FF) – zadanie B nie może zakończyć się dopóki nie zakończy się zadanie A,
- Start – Koniec (ang. Start-to-Finish SF) – zadanie B nie może zakończyć się dopóki nie rozpocznie się zadanie A.

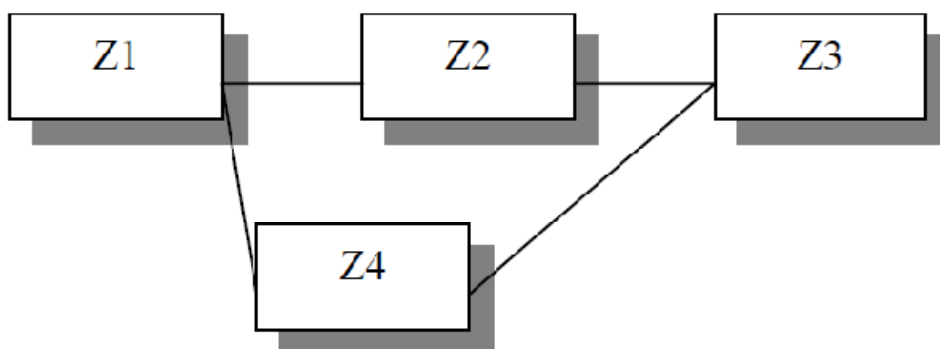


Rysunek 1: Typy powiązań między zadaniami w projekcie

Standardowo przyjmuje się, że zadania rozpoczynają się, gdy tylko jest to możliwe.

Zadania, których liczba w projekcie zwykle jest znaczna i tworzą harmonogram, mają takie atrybuty, jak:

- sekwencja,
- powiązanie,
- nakładanie się,
- ograniczenia (czasowe), data startu i zakończenia.

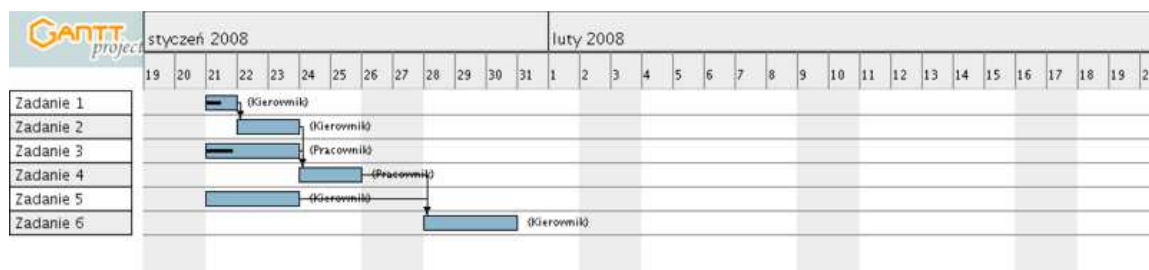


Rysunek 2: Przykład zadań Z1, Z2, Z3, Z4, z których składa się projekt

Zadania Z2 i Z3 są realizowane sekwencyjnie po wykonaniu zadania Z1, a zadanie Z3 po zrealizowaniu zadania Z2 i Z4. Z takiego graficznego przedstawienia zadań jak na rys. 2.7 nic nie możemy wnioskować o ograniczeniach czasowych ani o oczekiwanych zasobach przewidzianych do ich realizacji.

## 2. WYKRES GANTTA

Opracowany przez siebie wykres H. L. Gantt po raz pierwszy zastosował do przedstawienia planu produkcji formę graficzną w roku 1917. Na typowym wykresie Gantta wiersze zawierają stanowiska pracy, natomiast kolumny oznaczają jednostki czasu. Układ zdarzeń na wykresie przedstawiany jest najczęściej w wersji planowanej przed rozpoczęciem działania oraz rzeczywistej nanoszonej na wykres wraz z upływem czasu. Za pomocą wykresu Gantta można nie tylko planować i kontrolować wykonanie planu, ale także poprzez zastosowanie odpowiedniego systemu oznaczeń uwzględniać zmienność przebiegu wykonania zadania.



Rysunek 1: Przykładowy wykres Gantta

W przypadku wykresu wydajności pracy analizę przeprowadza się na podstawie zapisów rzeczywistych uzupełnionych o oznaczenia zakłóceń (np. brak pracownika, brak materiałów, brak instrukcji, remont maszyny, brak energii, brak narzędzi, brak doświadczenia wykonawcy, święta, narady, strajki, braki w kwalifikacjach wykonawcy, brak zlecenia).

Wykres Gantta jest graficznym sposobem planowania i kontroli. Planowanie i koordynowanie przebiegu różnych czynności w przekroju czasowym odgrywa istotną rolę w tworzeniu i funkcjonowaniu organizacji. Wykresy Gantta służą do planowania działań wielopodmiotowych zarówno zespołowych, jak i grupowych. Przedstawiają następstwo kolejnych zdarzeń, uwzględniając również zadania wykonywane równolegle. Dzięki tej technice można także kontrolować realizację zaplanowanego przedsięwzięcia.

### 2.1. ETAPY

Technika ta obejmuje następujące etapy:

**I etap:** rozłożenie przedsięwzięcia na cele etapowe lub cele szczegółowe,

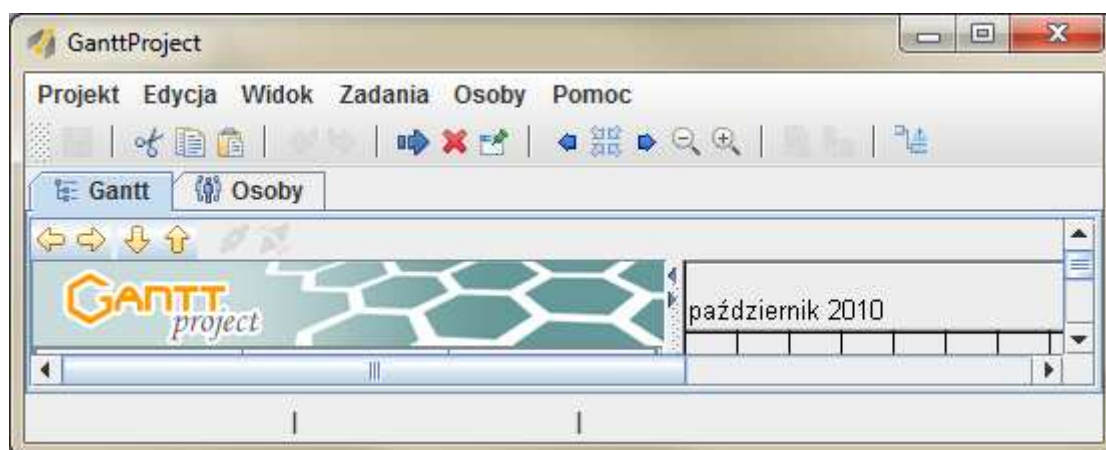
**II etap:** ustalenie czasu trwania przedsięwzięcia i określenie czasów realizacji celów etapowych i częściowych,

**III etap:** ustalenie kolejności realizacji celów etapowych i częściowych oraz wyznaczenie terminów ich rozpoczęcia i zakończenia,

**IV etap:** określenie miejsca, w którym cele te mają być zrealizowane,

**V etap:** wyrażenie w postaci graficznej wszystkich dokonanych czynności.

## 2.2. GANTTPROJECT - FUNKCJE MENU



Rysunek 2: GanttProject

**Projekt** - Otwieranie i zamykanie nowych i istniejących plików, zapisywanie i drukowanie plików, importowanie i eksportowanie projektu do wybranego formatu, m.in. html, pliki graficzne png, jpg, dokumenty pdf i csv, zapisywanie i otwieranie projektu z określonego serwera w sieci

**Edycja** - wycinanie, kopiowanie i wklejanie tekstu lub obiektu; ustawienia programu

**Zadania** – nowe zadanie, usuń zadanie; właściwości zadania

**Osoby** – nowa osoba, usuń osobę, właściwości osoby, wysyłanie osobom wiadomości elektronicznych, import osób z innego projektu (pliku gan, xml)

**Pomoc** - funkcje pomocy

## 2.3. IMPORT ZADAŃ

GanttProject potrafi wczytywać poza własnymi danymi także pliki w formacie MS Project oraz zwykłe pliki tekstowe.

### PRZYKŁAD NR 1 DO REALIZACJI

- Utwórz za pomocą notatnika plik tekstowy z następującą zawartością:

**Opracowanie koncepcji działania algorytmu**

**Utworzenie zadania projektowego**

**Implementacja algorytmu**  
**Testowanie algorytmu**  
**Implementacja poprawek po testowaniu**  
**Weryfikacja poprawności działania algorytmu**  
**Testowanie końcowe**  
**Implementacja poprawek po testowaniu końcowym**  
**Końcowa weryfikacja poprawności działania algorytmu**  
**Wdrożenie algorytmu**

- zapisz go następnie na lokalnym dysku w postaci pliku tekstowego (\*.txt),
- uruchom program GanttProject
- otwórz w nim zapisany wcześniej plik tekstowy (Projekt - Importuj - Pliki tekstowe \*.txt - Następna - wybierz plik z dysku - Akceptuj),
- zapisz uzyskany projekt na dysku (Projekt - Zapisz jako).

## 2.4. DODAWANIE RĘCZNE ZADANIA

Dodaj nowe zadanie do listy:

- Z menu programu (Zadania - Nowe zadanie),
- Myszą, za pomocą ikony ze strzałką (Nowe zadanie),
- Skrótom klawiszowym (CTRL+T),
- Z menu kontekstowego (prawy klawisz myszy).

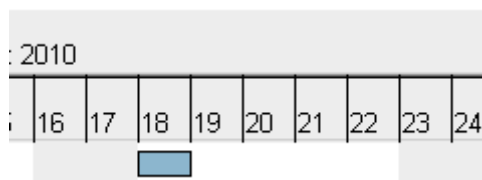
## 2.5. WŁAŚCIWOŚCI ZADANIA

- Kliknij prawym klawiszem myszy w pasek dowolnego zadania.
- Wybierz funkcję Właściwości zadania.
- Zbadaj zawartość wszystkich widocznych zakładek.
- Zauważ, że przypisanie niektórych parametrów wymaga wcześniejszego zdefiniowania w innym miejscu programu listy obiektów do wyboru (np. osoby).
- Zauważ, że lista możliwych parametrów jest otwarta - można dodawać, definiować i opisywać nowe kolumny danych, stosownie do potrzeb.

## 2.6. MODYFIKACJA PRAMETRÓW CZASOWYCH ZADANIA

### PRZYKŁAD 2 DOZREALIZOWANIA

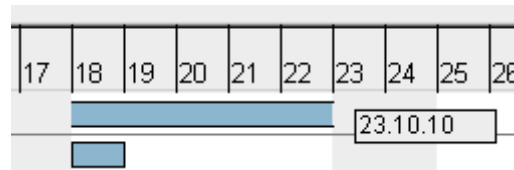
- Przeciągnij myszą oś czasu tak, aby odsłonić początki zadań.



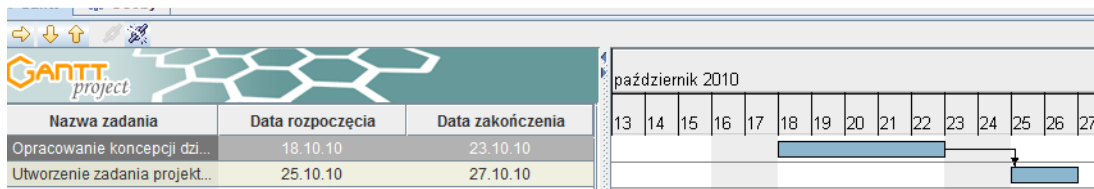
- Przesuń oś czasu za pomocą niebieskiej strzałki.



- Chwyć myszą prawą krawędź zadania i przeciągnij w poziomie, aby zmienić czas jego trwania (i jednocześnie czas zakończenia).



- Chwyć myszą lewą krawędź zadania i przeciągnij, aby zmienić dzień jego rozpoczęcia.
- Chwyć myszą lewą krawędź zadania nieco niżej (powinien się pojawić czarny kursor ze znakiem %) i przeciągnij w prawo, aby oznaczyć jego postęp (stan realizacji) w procentach. W ten sposób będzie można prosto oznaczać zadania zrealizowane.
- Chwyć myszą pasek zadania i przeciągnij aż zobaczysz cienką strzałkę. Puść przycisk myszy nad paskiem sąsiedniego zadania. Zadania zostały powiązane strzałką symbolizującą relację koniec-początek. Od tego momentu zakończenie jednego z nich automatycznie warunkuje rozpoczęcie drugiego. Relację można usunąć w menu kontekstowym zadania (prawy przycisk myszy na pasku zadania) funkcją Rozdziel zadania.



- Kliknij w datę rozpoczęcia lub zakończenia zadania i zmień ręcznie datę.
- Kliknij dwukrotnie w pasek zadania i sprawdź, jakie jeszcze inne parametry można do niego przypisać.

## 2.7. ROLA W PROJEKCIE

**Executive Sponsor (Project Champion)** - rola, mająca możliwość i uprawnienia zatwierdzać zmiany w projekcie oraz zdobywać potrzebne zasoby, materiały; ma decydujący głos w dyskusjach; sponsor projektu.

**Visionary** - osoba odpowiedzialna za to aby w miarę wcześnie sprecyzować wymagania, posiada najlepsze rozeznanie w dziedzinie zastosowań produkowanego systemu, nadaje projektowi kierunek rozwoju, często pomysłodawca systemu.

**Ambassador User** - odpowiedzialny za przekazanie wiedzy od klienta, odpowiedzialny jest za całokształt projektu, zapewnia sprzężenie zwrotne programistom podczas implementacji.

**Advisor User** - rolę tę pełni użytkownik (użytkownicy) wyznaczeni do reprezentowania danego punktu widzenia na projekt, są przedstawicielami użytkowników, ma wgląd w realizowany projekt a w miarę potrzeb potrafi dostarczyć programistom odpowiedniej wiedzy na dany temat - za który jest odpowiedzialny.

**Technical Co-ordinator** - osoba odpowiedzialna za architekturę systemu, kontrolę realizacji oraz techniczną jakość projektu, odpowiedzialny za architekturę i jakość produktu, zarządza zmianami w projekcie.

**Team Leader** - dowodzi zespołem ludzi i zapewnia im stale możliwość efektywnej pracy.

**Developer** - implementuje system, interpretuje wymagania systemowe oraz model systemu, dostarcza kod programu i buduje prototypy.



**Senior Developers** - odpowiedzialni za analizę, starsi programiści wyznaczani są na podstawie doświadczenia w danej dziedzinie lub technologii.

**Tester** - testuje poprawność techniczną produktu, pisze testy, dodaje odpowiednie komentarze i tworzy dokumentację.

**Scribe** - odpowiedzialny za zbieranie i zapisywanie wymagań, porozumień i decyzji podjętych przy wspólnej pracy

**Facilitator** - kieruje postęпами prac, stanowi motor do przygotowań do pracy a także jest odpowiedzialny za dobry przebieg komunikacji

**Business Architect, Quality Manager, System Integrator** - osoby pełniące inne funkcje podczas realizacji projektów tradycyjnymi metodami

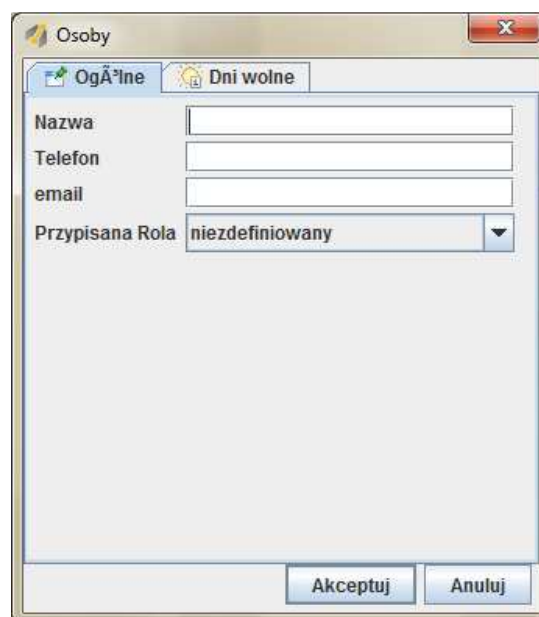
## 2.8. OSOBY I ROLA W PROJEKCIE

W procesie planowania projektów po ustaleniu listy zadań, ich parametrów oraz zależności pomiędzy zadaniami przychodzi pora na przydzielenie zasobów (w tym tzw. zasobów ludzkich). Dla pojedynczej realizacji niewielkiego projektu mogą to być konkretne osoby fizyczne (reprezentowanymi przez imię, nazwisko, dane kontaktowe). Istotne jest, jakie role poszczególni uczestnicy spełniają w danym projekcie. Dopiero w następnej kolejności konkretnym osobom o ustalonych rolach przypisywać będziemy określone zadania do realizacji zgodnie z planem projektu.

### DODAWANIE OSOBY DO PROJEKTU

Dodaj do projektu osobę kilkoma sposobami:

- Z menu programu (Osoby - Nowa osoba,
- Skrótem klawiszowym (CTRL+H).



Wybierz na głównym ekranie zakładkę Osoby (domyślnie była wybrana zakładka Gantt), a następnie wypróbuj dwie następne metody dodawania osób:

- z menu kontekstowego - prawy klawisz myszy,
- myszą, za pomocą ikony ze strzałką (Nowa osoba).



---

## NIESTANDARDOWE ROLE

Role niestandardowe definiuje się globalnie, dla potrzeb prowadzonych przez daną instytucję projektów.

- Wejdź do konfiguracji programu (Edycja - Parametry - Rola osoby).
- Dodaj przykładowe role nauczyciel i uczeń (klikając dwukrotnie w wolne linie tabeli).
- Zamknij okno konfiguracji.
- Wybierz właściwości osoby (dowolnej). Sprawdź, czy na liście dostępnych ról pojawiły się nowe, wprowadzone przez Ciebie.

---

## PRZYKŁAD NR 2 DO REALIZACJI

- Zdefiniować wszystkich użytkowników i wszystkie role biorące udział w przykładowym projektem
- Dodać relacje do wszystkich zadań w projekcie

---

## PRZYKŁAD 3 DO REALIZACJI

- Opracować kompletny harmonogram przedsięwzięcia informatycznego – tworzenia oprogramowania (minimum 10 zadań, 10 osób, co najmniej 3 programistów, co najmniej 2 miesiące)