

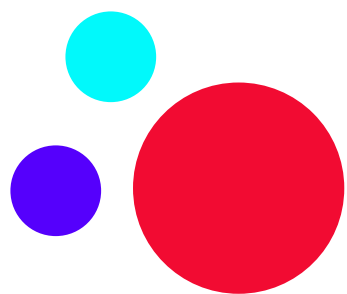
# PRACA W ZESPOLE: JIRA

infoShare Academy



# PRACA W ZESPOLE: JIRA

infoShare Academy



## **Łukasz Koral**

Pomagam programistom budować wydajne zespoły i osiągać w nich fantastyczne wyniki.



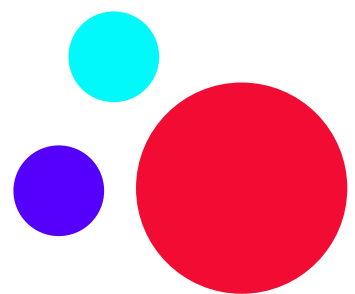


# Agenda

Czyli o czym będziemy dzisiaj rozmawiać?

- Wprowadzenie do JIRA.
- Praca w JIRA.
- Zasady skutecznej komunikacji.
- Podział pracy w zespole.
- Co programista powinien robić poza pisanie kodu?



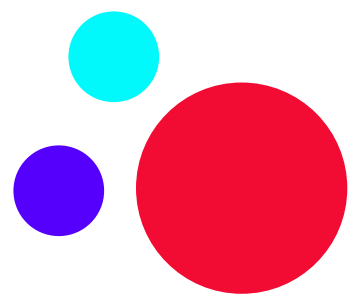


# Wprowadzenie do JIRA

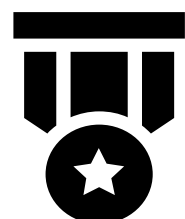


# Jira Software

info **Share**  
ACADEMY



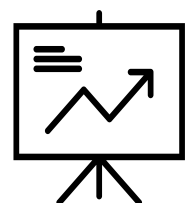
# Wprowadzenie do JIRA



Jeden z **najpopularniejszych** systemów do zarządzania pracami projektowymi.



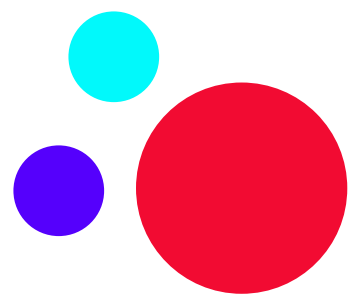
Używane przez **ponad** 100 000 organizacji na całym świecie.



Wyznacza **standardy** pracy w systemach zarządzania projektami.

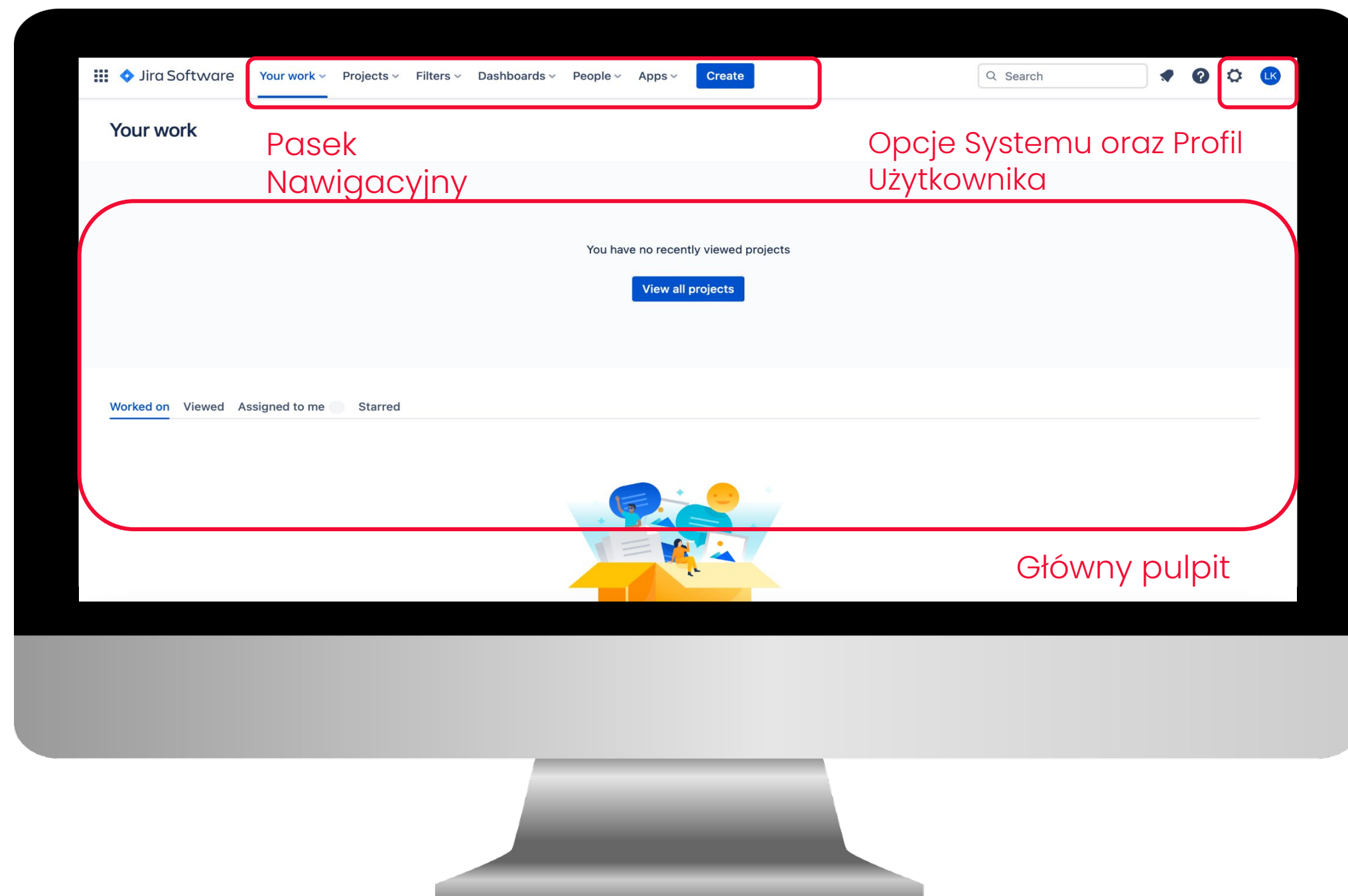
info **Share**  
ACADEMY

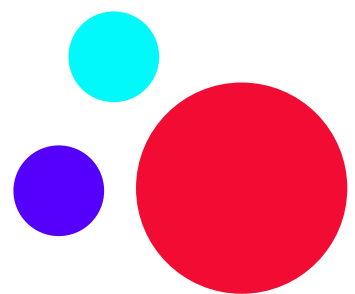




# Wprowadzenie do JIRA

infoShare  
ACADEMY





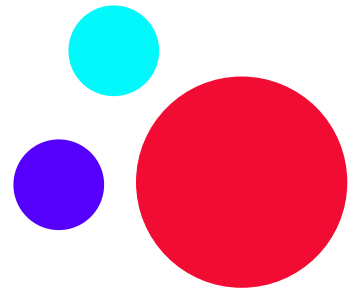
**Praca w JIRA**

**infoShare**  
ACADEMY



# Jira Software

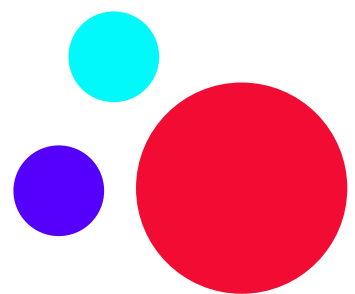




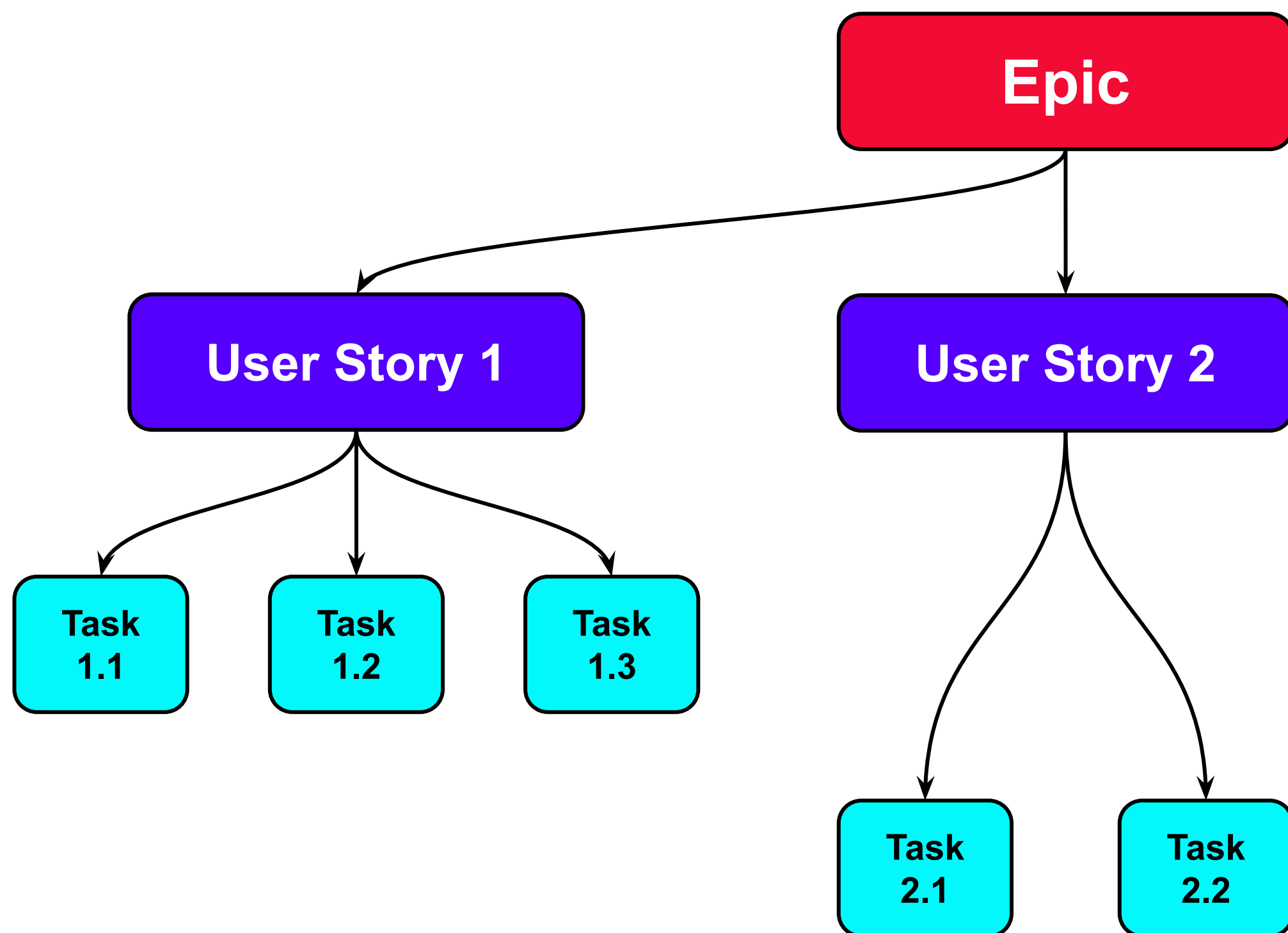
# Ćwiczenie 1 (instrukcja)

Tworzenie projektu:

- Template Kanban.
- Template Scrum.



# Praca z zadaniami







## Ćwiczenie 2 (instrukcja)

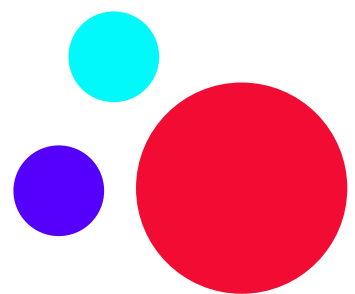
Tworzenie zadań:

→ Tworzenie zadań:

- Epic (1),
- Story (2),
- Subtask (3/Story).

→ Powiązanie ze sobą już istniejących zadań.

infoShare  
ACADEMY



## Ćwiczenie 3 (instrukcja)

Praca z zadaniami – podstawowe zagadnienia:

- Przypisywanie zadań do użytkowników.
- Zmiana statusów zadań.
- Praca z komentarzami.
- Zmiana parametrów zadań:
  - Priorytet,
  - Label,
  - Wersja,
  - Sprint.

**infoShare**  
ACADEMY



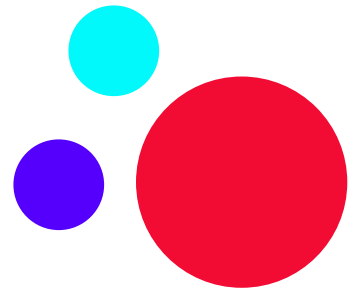


## Ćwiczenie 4 (instrukcja)

Nawigacja pomiędzy widokami:

- Backlog.
- Bieżący Sprint (template Scrum).
- Kanban Board (template Kanban).
- Releasey.





## Ćwiczenie 5 (instrukcja)

Wyszukiwanie zadań:

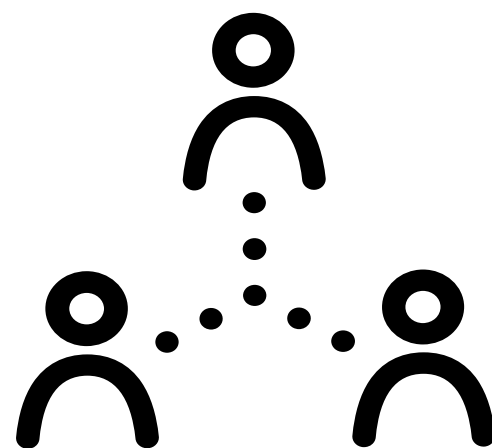
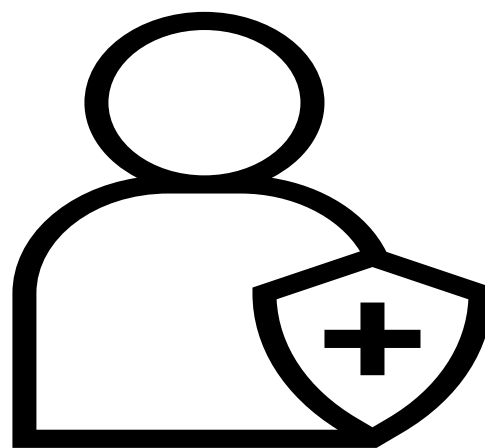
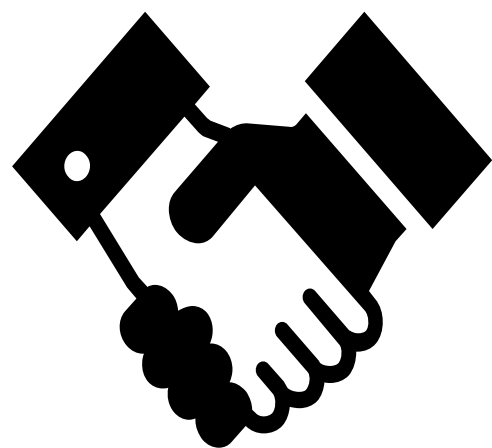
- Używanie silnika wyszukiwania zadań w Jira w wersji podstawowej.
- Zaawansowane wyszukiwanie za pomocą JQL.



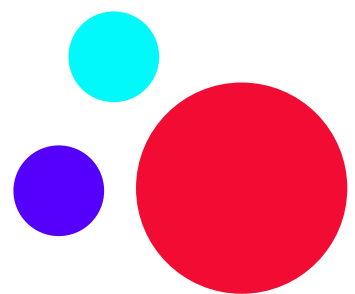




# Zasady **skutecznej** komunikacji



**info** Share  
ACADEMY



# Zasady **skutecznej** komunikacji

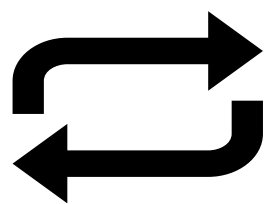
**“Nie ma czegoś takiego jak za dużo komunikacji.”**

Autor nieznany

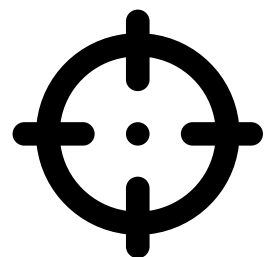




# Zasady **skutecznej** komunikacji



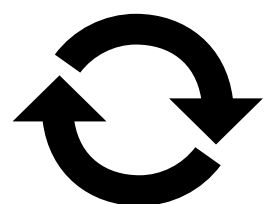
Zwracaj uwagę na jakość komunikacji **zarówno** wewnątrz jak i na zewnątrz zespołu deweloperskiego.



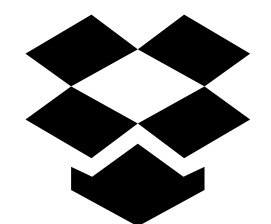
Dbaj o **merytorykę** wypowiedzi. Mów o faktach i konkretach – unikaj skrótów myślowych i uogólnień.



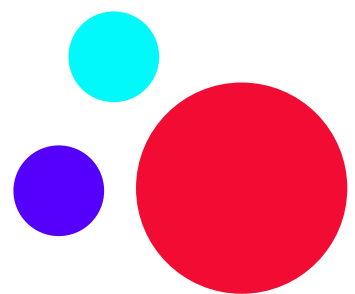
Jeśli to możliwe **dostosuj** język wypowiedzi do odbiorcy.



Jasno komunikuj **skutki i konsekwencje** decyzji jeśli prezentujesz możliwe rozwiązania.



Zachowaj **otwartą** komunikację. Mów o swoich oczekiwaniach i uczuciach względem sytuacji, nie unikaj mówienia o problemach.



## Klątwa wiedzy



Klątwa wiedzy to **zjawisko** występujące u osób posiadających określony zasób wiedzy w jakiejś dziedzinie.



Polega ono na tym, że dana osoba **traktuje informacje jako oczywiste** i prowadząc rozmowę zakłada, że druga strona również je posiada lub rozumie.

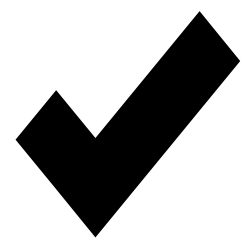


W efekcie podczas komunikacji **obie strony mogą zakładać złe wzajemne intencje** – nadawca brak chęci przyswojenia informacji przez odbiorcę a odbiorca brak chęci przekazania informacji przez nadawcę.

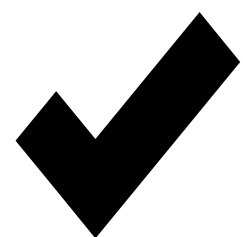




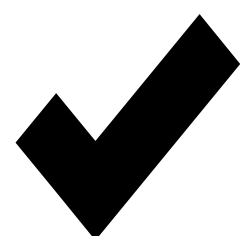
## Jak **unikać** klątwy wiedzy?



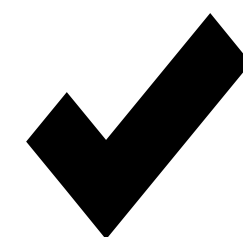
Nigdy **nie zakładaj**, że odbiorca jest zaznajomiony z jakąś informacją.



Komunikując się z osobą nie posiadającą wiedzy domenowej **tłumacz pojęcia prostymi słowami**, używaj prostych konstrukcji zdań.



Przedstawiając koncepcje zawsze **tłumacz lub rozwijaj** pojęcia, nawet te, które Tobie wydają się oczywiste.

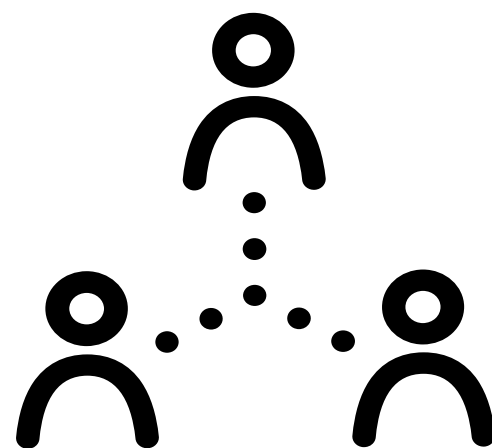
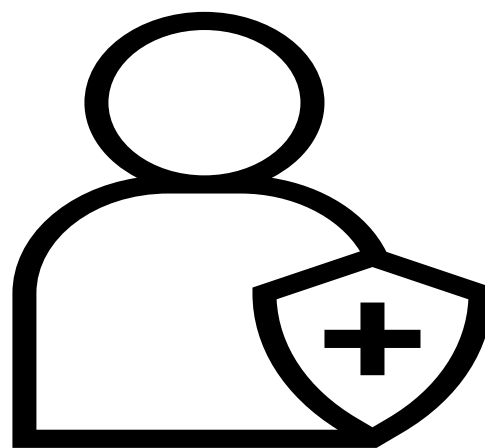
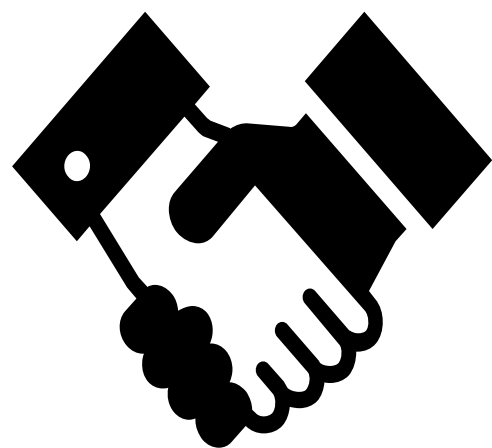


Zawsze zakładaj dobre intencje odbiorcy wiadomości. **Jeśli masz wątpliwości zapytaj**, czy przekaz jest jasny.

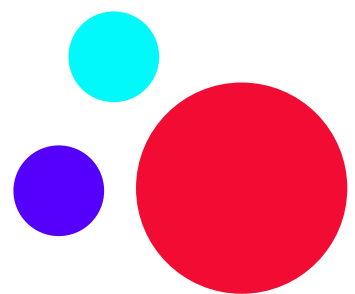




# Podział pracy w zespole

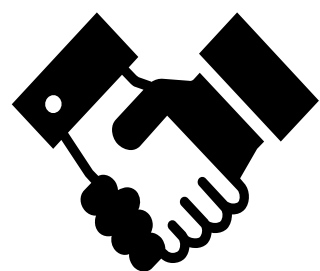


infoShare  
ACADEMY

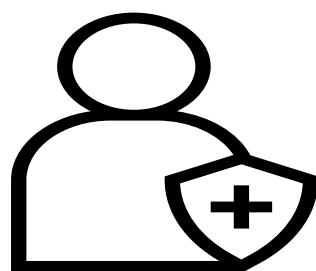


# Podział **pracy** w zespole

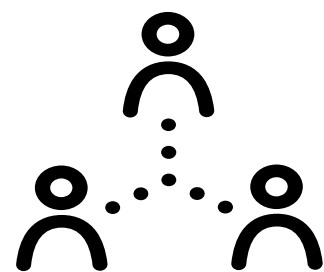
**Samoorganizacja**

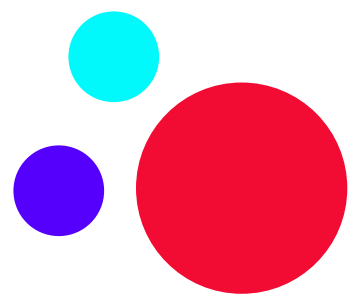


**Odpowiedzialność**



**Zaangażowanie**





# Co programista powinien robić poza pisananiem kodu?





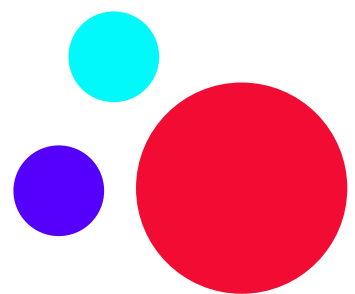


**Nie tylko pisanie kodu**

**Big Picture**

**Sfera biznesowa  
projektu**

**Dzielenie się  
wiedzą i  
informacjami**



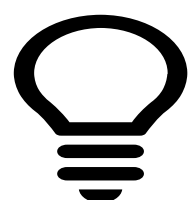
# Nie tylko pisanie kodu – **Big Picture**



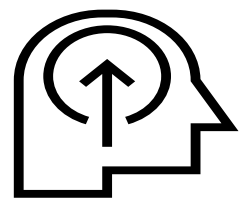
Projekt to **nie tylko** funkcjonalność lub moduł nad którym pracujesz.



Świadomość **pozostałych** części systemu nad którym pracujesz pozwoli Ci podejmować lepsze decyzje.

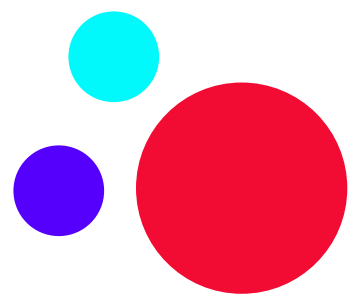


Często odpowiedzi na Twoje wątpliwości i pytania możesz znaleźć poza 'swoją' częścią systemu.

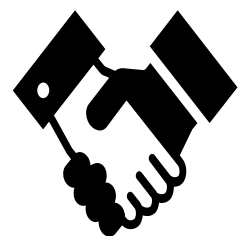


Szersza perspektywa sprawia, że jesteś **bardziej świadomym** członkiem zespołu.

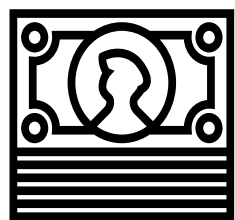
info **Share**  
ACADEMY



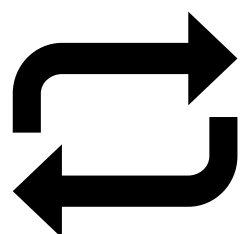
# Nie tylko pisanie kodu – **Sfera biznesowa**



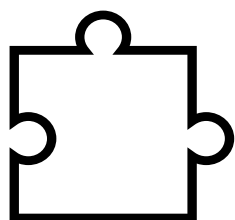
‘Biznes’ nie jest Twoim przeciwnikiem – **razem** Tworzycie produkt nad którym pracujesz.



Uwarunkowania i decyzje biznesowe **mają duży wpływ** na ‘techniczną’ część projektu.



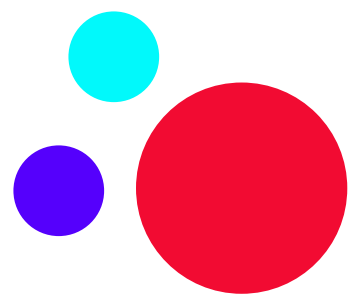
Często decyzje techniczne **mogą mieć** wpływ na część biznesową projektu.



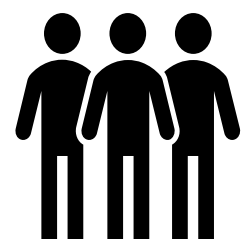
Łatwiej zrozumiesz argumentację decyzji i uwarunkowań mając świadomość **wszystkich** czynników na nie wpływających.

info **Share**  
ACADEMY

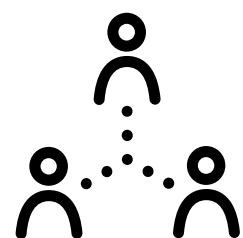




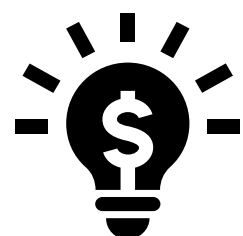
# Nie tylko pisanie kodu – Dzielenie się wiedzą



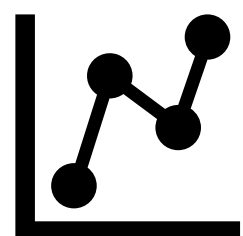
Oczekując pomocy od innych – sam również **powinieneś** być gotów jej udzielić.



Razem możecie **więcej**! Budowanie wspólnej bazy wiedzy w sprawie, że osiągniecie lepsze wyniki jako zespół.



Wiedza i informacje które posiadasz **mogą być** tym czego właśnie szuka kto w Twoim zespole.



Dzielenie się wiedzą to również element i forma **Twojego** rozwoju.

info **Share**  
ACADEMY

# Dziękuję bardzo za uwagę!

[infoShareAcademy.com](https://infoShareAcademy.com)