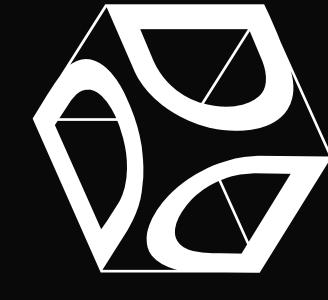



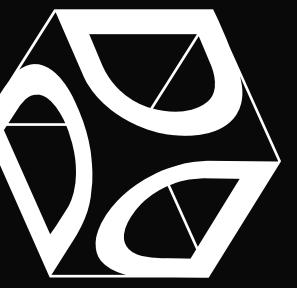



Lekcja 1

Nowe Wymagania Rzucające Wyzwanie Modelowi


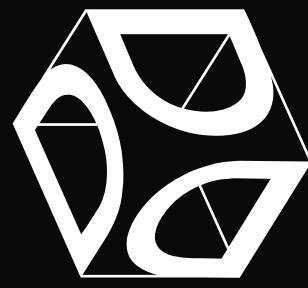


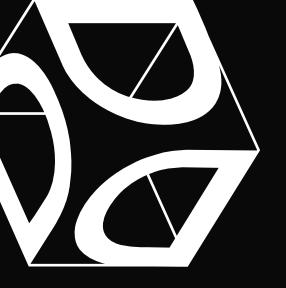
I tak to się z... epsuje



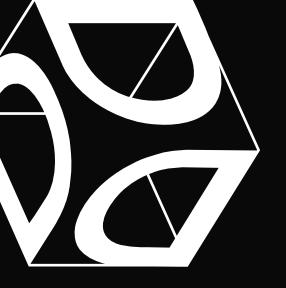


Pierwsze wymagania są tylko pierwszymi wymaganiami.





Modularyzacja zwiększa prawdopodobieństwo
bezstresowego otwierania nowych potencjałów
biznesowych



Modularyzacja zwiększa prawdopodobieństwo
bezstresowego otwierania nowych potencjałów
biznesowych



Modularyzacja zwiększa prawdopodobieństwo
bezstresowego otwierania nowych potencjałów
biznesowych

...ale nie eliminuje go.


Dwa rodzaje przetomowych wymagań

Dwa rodzaje przełomowych wymagań

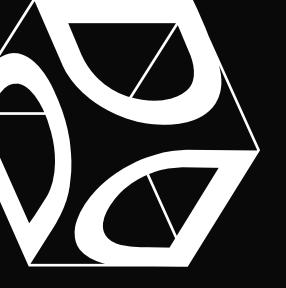
- przełomowe dla biznesu

Dwa rodzaje przełomowych wymagań

- przełomowe dla biznesu
- przełomowe (łamiące) dla architektury



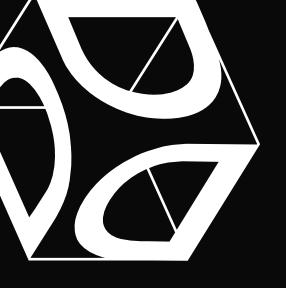
Plan gry



Zadanie świadomego programisty/programistki:


Dowiezienie tematu oraz wzięcie odpowiedzialności za swoje decyzje i ich argumentowanie (nawet jeśli tąmią jakieś ogólnie przyjęte dobre praktyki).

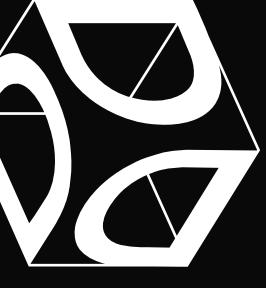
Zadanie



Niektóre zasoby mogą pełnić specyfczną rolę w wielu projektach jednocześnie.

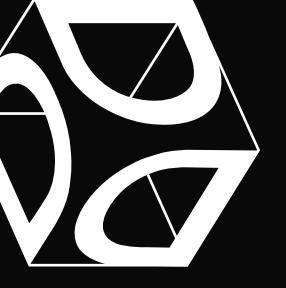
Oczywiście jeśli są niesprawne, to nie mogą być w żadnym projekcie.





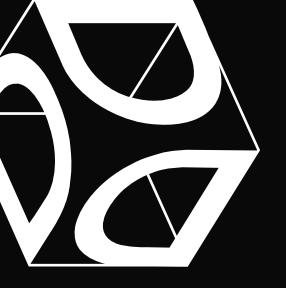
```
git checkout allocation-availability-done
```

Zadanie cd.



```
// ..availability module
boolean block(ResourceId resourceId, TimeSlot timeSlot, Owner requester) {
    ResourceGroupedAvailability toBlock = findGrouped(resourceId, timeSlot);
    return block(requester, toBlock);
}


// ..allocation module
Optional<UUID> allocateToProject(ProjectAllocationsId projectId, ResourceId resourceId, Capability capability, TimeSlot timeSlot) {
    // ..
    return event.map(CapabilitiesAllocated::allocatedCapabilityId);
}
```

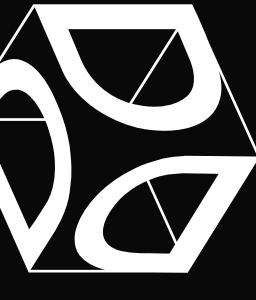


```
// ..availability module
boolean block(ResourceId resourceId, TimeSlot timeSlot, Owner requester) {
    ResourceGroupedAvailability toBlock = findGrouped(resourceId, timeSlot);
    return block(requester, toBlock);
}

// ..allocation module
Optional<UUID> allocateToProject(ProjectAllocationsId projectId, ResourceId resourceId, Capability capability, TimeSlot timeSlot) {
    // ..
    return event.map(CapabilitiesAllocated::allocatedCapabilityId);
}
```

Do tej pory można blokować dostępne zasoby i dodawać je do projektów, ale tylko wybierając konkretny zasób. Teraz chcemy zablokować dowolny zasób o konkretnych umiejętnościach.





```
git checkout allocation-availability-done
```

Zaproponowany
sposób zależy od:

Zaproponowany sposób zależy od:

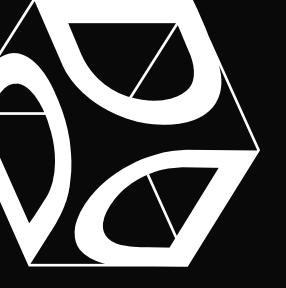
- dostępnego czasu

Zaproponowany sposób zależy od:

- dostępnego czasu
- topologii zespołów

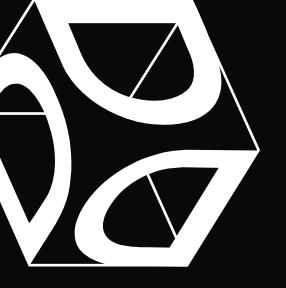


Pytania pomocnicze



Pytania pomocnicze

- Jak można byłoby to zrobić, gdyby czasu było więcej?



Pytania pomocnicze

- Jak można byłoby to zrobić, gdyby czasu było więcej?
- Jak można byłoby to zrobić, gdyby nie było problemu synchronizacji zespołów?