

Inżynieria wymagań i jakości



Organizacja przedmiotu



Organizacja

- Wykłady – prowadzone zdalnie za pomocą MS Teams, na kanale Ogólnym
 - Laboratoria – prowadzone stacjonarnie; ćwiczenia w małych grupach, realizacja na laboratoriach praktycznej analizy wymagań
 - Sesje online – wykorzystanie opcji czatu oraz pytań bezpośrednich (podnoszenie łapki)
 - Wszystkie materiały do przedmiotu na platformie UPEL:
<https://upel.agh.edu.pl/course/view.php?id=7173> (zapisy na laboratoriach)
 - Preferowany sposób komunikacji: MS Teams, poczta elektroniczna
-

Prowadzący



Kamil Piętak

kpietak@agh.edu.pl



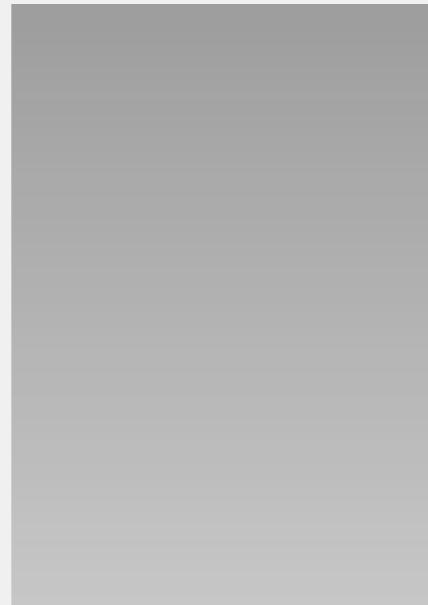
Jacek Dajda

dajda@agh.edu.pl



Maria Eckes-Kondak

meckes@agh.edu.pl



Daria Wójcik

dwojcik@agh.edu.pl

Zasady zaliczenia

- W trakcie ćwiczeń laboratoryjnych odbywać się będą krótkie kartkówki na rozpoczęcie zajęć, które podsumowują wiedzę z poprzedniego laboratorium. Kartkówki będą dotyczyć wiedzy z laboratorium od 1 do 4 (włącznie). Za kartkówkę będzie można otrzymać **0-8 pkt**. Łącznie za tą część zajęć można otrzymać **0-32 pkt**
- Za mini-projekt tworzony pomiędzy laboratorium 4-6 można otrzymać **0-18 pkt**
- Na koniec semestru odbędzie się **kolokwium** dotyczące zarówno teoretycznej wiedzy z wykładu, jak również praktycznych aspektów tworzenia wymagań systemów informatycznych; za kolokwium można otrzymać **0-50 pkt**
- Do uzyskania zaliczenia student musi otrzymać co najmniej 50% punktów z części laboratoryjnej (kartkówki i mini-projekt) oraz 50% punktów z kolokwium zaliczeniowego.
- W przypadku nieuzyskania wystarczającej liczby punktów student zobowiązany jest do napisania poprawkowego kolokwium zaliczeniowego i/lub przedstawienia poprawionej wersji pracy wykonywanej w ramach części laboratoryjnej.

Inżynieria wymagań i jakości





Inżynieria wymagań w praktyce

Źródło:

<https://twitter.com/riferrei/status/1306915448382984192>

Software Requirement Specifications



Interesariusze w procesie budowy oprogramowania



Biznes

Chce zarobić.



Klient

Stara się dostać rozwiązanie, które pomoże mu rozwiązać określone problemy. Często poruszający się w określonej dziedzinie / ekosystemie, która/e nie musi być zrozumiała dla innych.



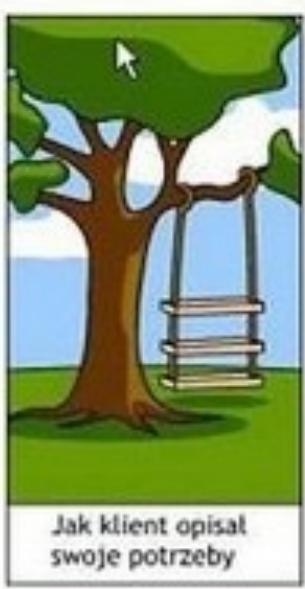
Analityk systemowy

Jej/jego zadaniem jest odkrycie potrzeb klienta, wyspecyfikowanie wymagań i przekazanie ich zespołowi.



Programista

Główny zadaniem jest zbudowanie produktu dopasowanego do potrzeb klienta. : Niekoniecznie oznacza to zrobienie wydajnego, najbardziej przemyślanego produktu w nowej technologii.



Jak klient opisał swoje potrzeby



Jak to zrozumiał projektant



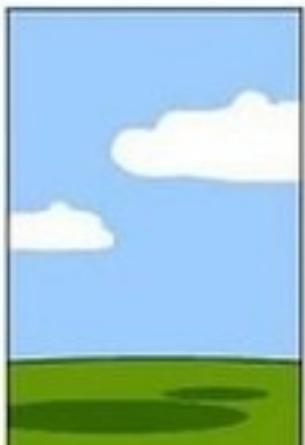
Jak to wymodelował analityk



Jak to programista zakodował



Jak to marketer przedstawił



Jak to zostało opisane w dokumentacji



Jakie funkcje zostały przewidziane



Za co zapłacił klient



Czego dotyczy pomoc techniczna



Czego klient naprawdę potrzebował

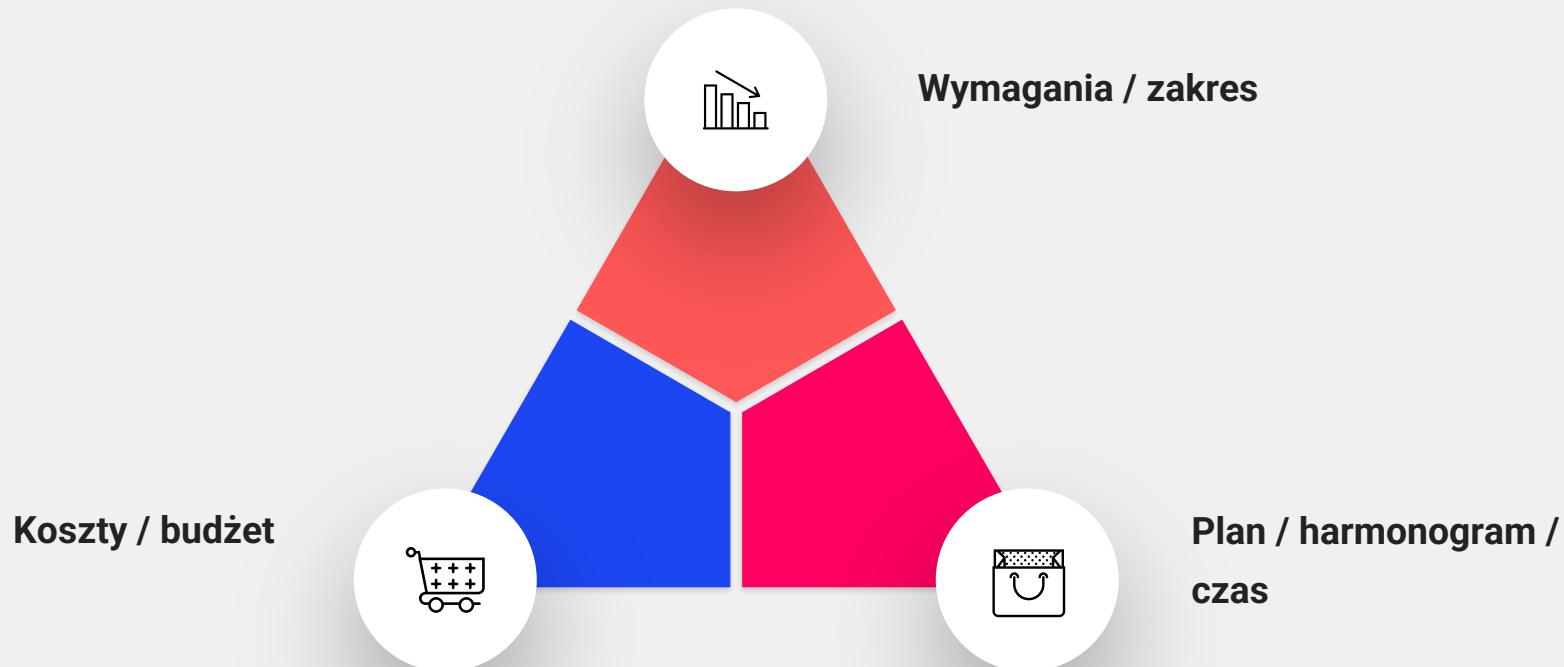
Wyzwania w procesie analizy wymagań

Na czym polega trudność/wyzwanie w analizie wymagań wobec systemu informatycznego?

- Zrozumienie realnych potrzeb użytkowników
- Identyfikacja problemów, które ma rozwiązywać system informatyczny
- Zrozumienie kontekstu funkcjonowania systemu, często zrozumienie innej/obcej branży / dziedziny
- Komunikacja z klientem / użytkownikiem oraz zespołem wykonawczym (w tym programistami)

Metody wytwarzania oprogramowania

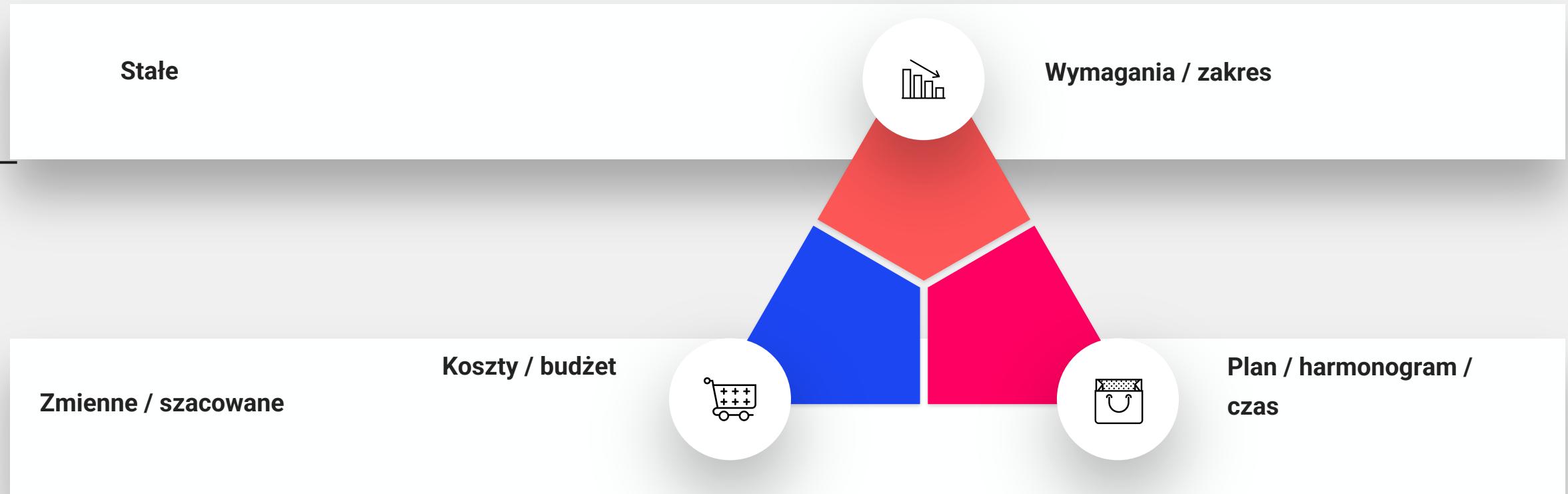
The iron triangle



Uznane metody wytwarzania oprogramowania

- Modele predyktywne: model kaskadowy
 - Metody iteracyjne: spiralny, RUP
 - Metody adaptacyjne: Agile
-

Model kaskadowy



Model kaskadowy



Model **kaskadowy**

kaskadowego

Implementacja

Implementacja rozwiązania
zgodnie z projektem

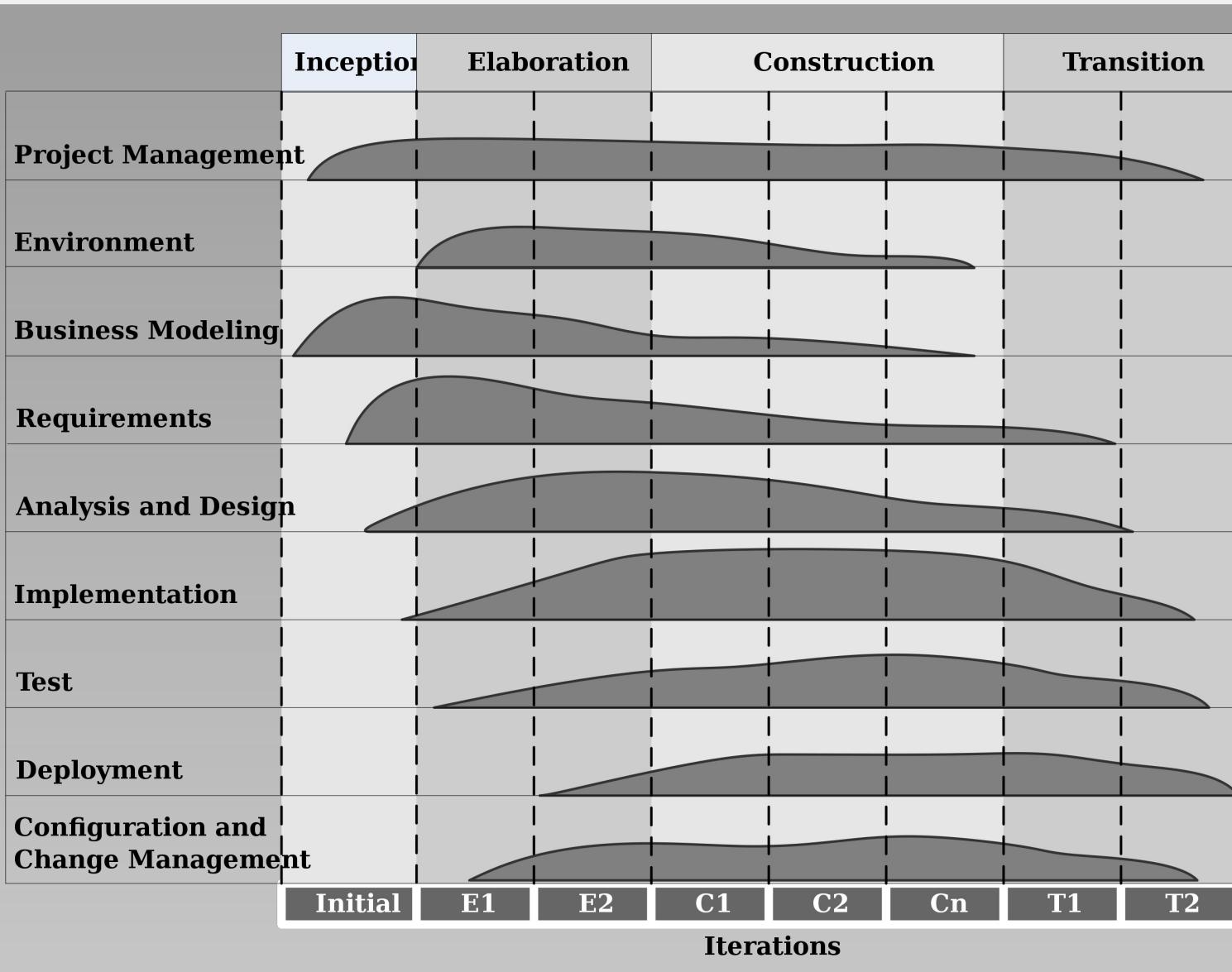
Testowanie

Testowanie zrealizowanych
komponentów systemu,
wprowadzanie ew. poprawek.

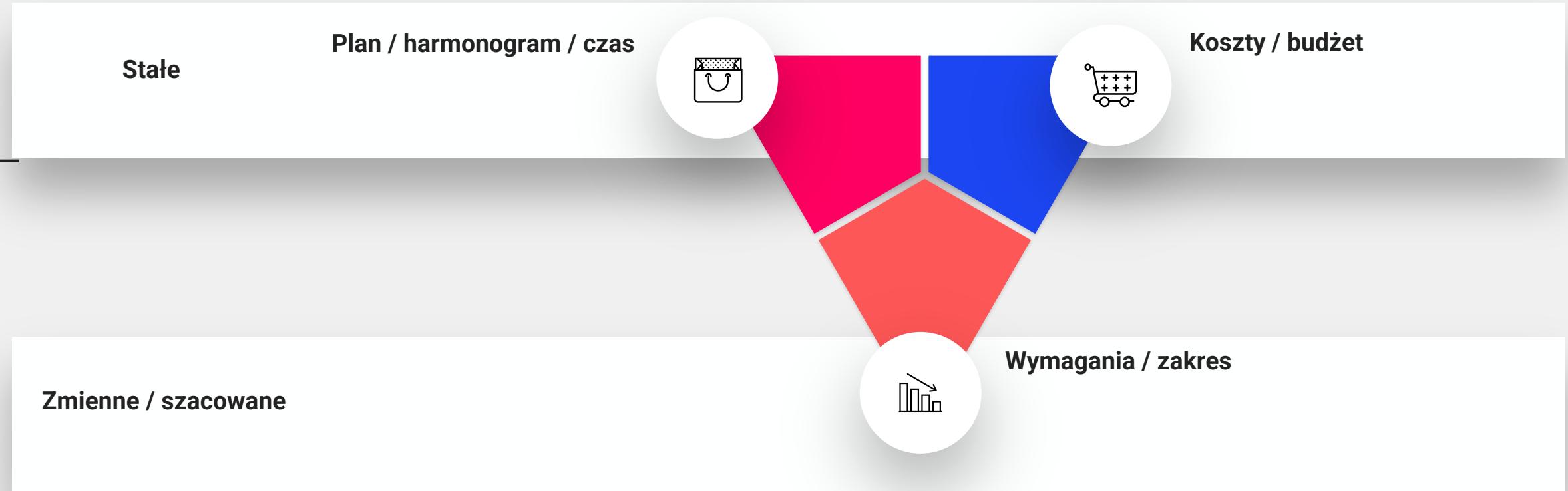
**Wdrożenie
Utrzymanie**

Dokumentowanie systemu
(użytkowe, techniczne),
wdrożenie i uruchomienie

Metody iteracyjne Rational Unified Process



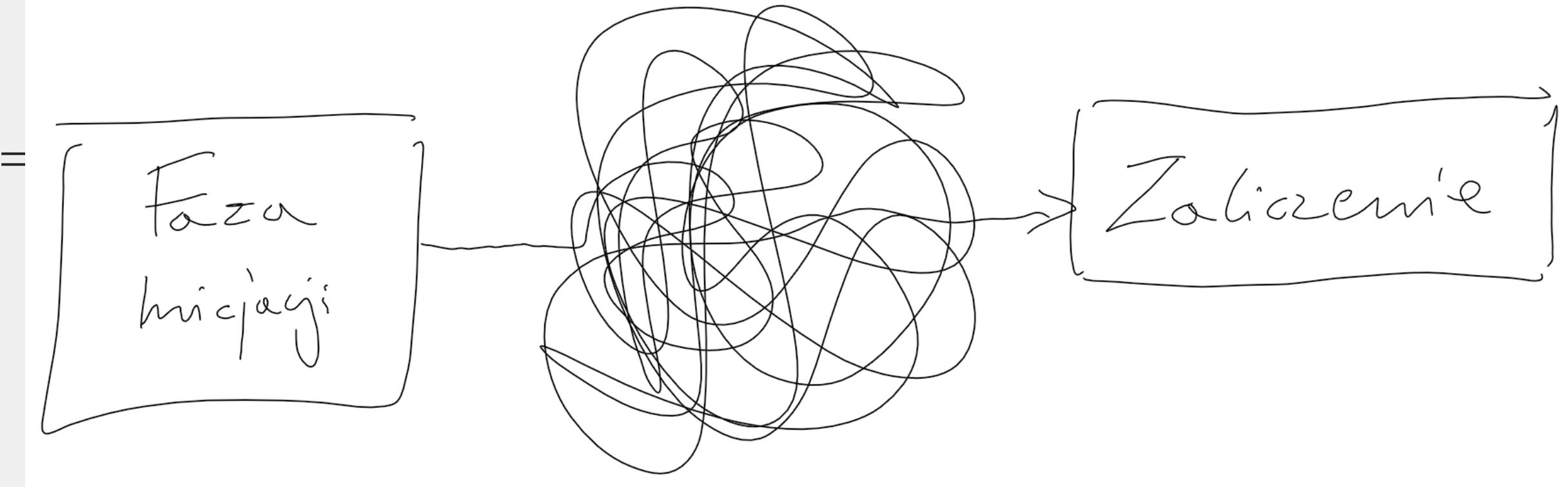
Agile



Metody zwinne – Agile

- Proces iteracyjno-ewolucyjny
- Podejście empiryczne: reagowanie na zmianę zamiast podążanie za planem
- Stałe terminy, zmienny zakres wymagań
- Ścisła współpraca pomiędzy wykonawcą a zamawiającym - bez tego niemożliwe jest precyzowanie wymagań i ich walidacja
- Nacisk na działające oprogramowanie zamiast dokumentacji

Model studencki



Inżynieria wymagań a programiści

- Przecież powiedzą mi dokładnie co mam zrobić...?
- Programiści pracują z wymaganiami – dostają je "na wejściu", na ich podstawie tworzą oprogramowanie
- Muszą więc je dobrze rozumieć i umieć je weryfikować w kontekście potencjalnych błędów
- W procesach mniej sformalizowanych wymagania są mniej doprecyzowane, a "klient" bliżej. Wtedy programiści biorą udział w ich formowaniu

Inżynieria wymagań a programiści

- Jako programiści wykorzystujemy zewnętrzne oprogramowanie, a czasami zamawiamy je (outsourcing). Wtedy my stajemy się klientami
- Jeśli marzy Ci się praca nad własnym start-up'em: będziesz musiał tworzyć wizje i opracowywać wymagania
- Żeby zweryfikować jakość zbudowanego oprogramowania musimy dobrze rozumieć wymagania, by na ich podstawie testować to co budujemy

Inżynieria wymagań a programiści

- Problem "Klienta z wizją"
 - Często ogólne hasła
 - Brak przemyślanych wszystkich funkcji
 - Brak doprecyzowania częstych scenariuszy
 - Tylko pozytywne scenariusze
- Siłą rzeczy błędy i braki koncepcyjne dotrą do zespołu i zespół je musi zidentyfikować. Im wcześniej tym lepiej (oszczędność czasu poświęconego na programowanie i testowanie)

Proces analizy wymagań

Proces analizy wymagań

Analiza od ogółu do szczegółu

W procesie analizy wymagań wyróżniamy kroki i narzędzia umożliwiające modelowanie wymagań na różnym poziomie szczegółowości.

Diagram obok wyróżnia wybrane elementy, które można wybrać w procesie modelowania wymagań. Ich użycie powinno być dostosowane do potrzeb danego projektu.

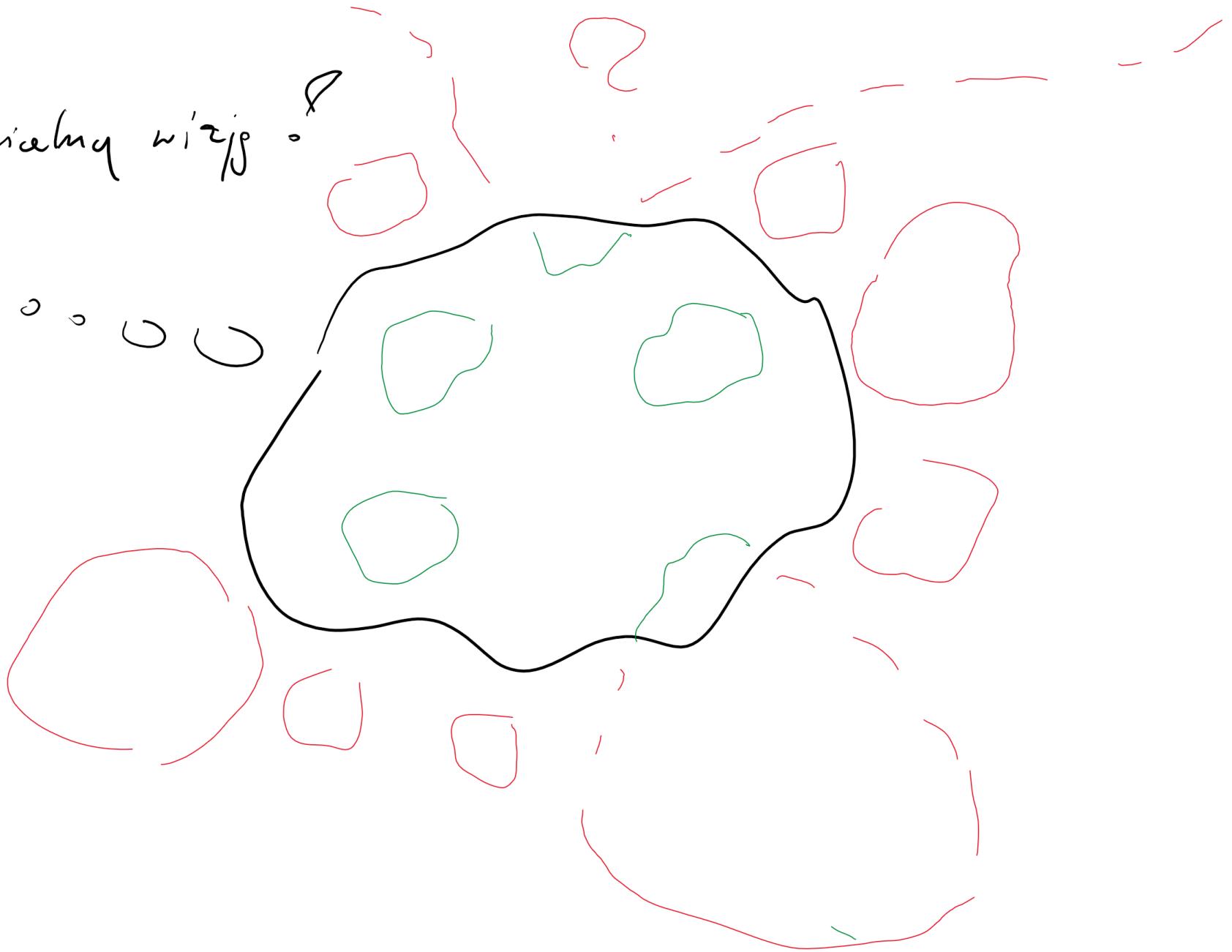


=

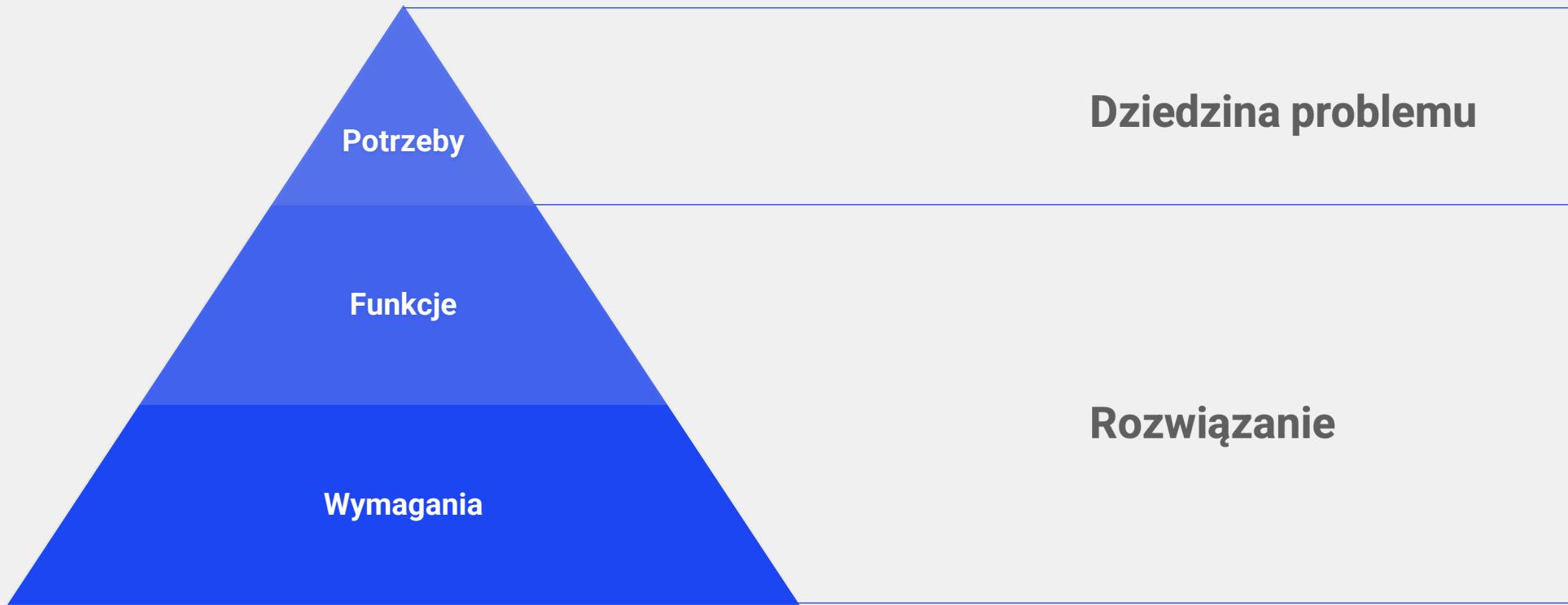
Od czego zacząć projekt?



Mom genneting wizj.

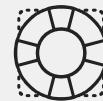


Piramida wymagań





Start-up



Co?

Czym będzie produkt który wymyśliliśmy? Aplikacja mobilna, webowa, zintegrowany system czujników, autonomiczny robot?



Dla kogo?

Kto będzie korzystał z proponowanego rozwiązania?



Dlaczego?

Dlaczego "ten ktoś" miałby skorzystać z naszego rozwiązania? Jakie problemu pozwoli mu rozwiązać? Jakie korzyści mu to przyniesie?



Zamówienie od klienta



Mam problem, chcę coś usprawnić

Zazwyczaj punktem wyjścia do nowego systemu jest chęć rozwiązania problemu, usprawnienia działania biznesu i osiągnięcia pewnych korzyści



Inni też takie mają, ja też chcę

Czasem klienci chcą mieć też dane rozwiązanie bo inni mają takie samo.



Chcę spełnić wymogi prawne

Konieczność posiadania niektórych rozwiązań wynika z funkcjonujących przepisów prawa (np. kasy fiskalne, programy do fakturowania)

Wizja

Zawiera odpowiedzi na pytania

Odpowiedzi na postawione obok pytania to nic innego jak wizja projektu. Wizja może przyjmować różne formy, bardziej lub mniej formalne. Zawsze jednak powinna dotyczyć wspomnianych aspektów i klarownie je wyjaśniać.

Wizja kierowana jest zarówno na zewnątrz (do klienta lub inwestora), jak również do wewnętrz (do zespołu wykonawczego). Jest to pierwszy krok do wspólnego zrozumienia celów projektu, na razie na bardzo wysokim poziomie abstrakcji.

Jaka jest dziedzina problemu?

W jakim obszarze będzie działał docelowy system? Jak wygląda obecna sytuacja w instytucji, która będzie wdrażała produkt. Wyjaśnienie kontekstu potrzebnego do zrozumienia problemów zarysowanych poniżej.

Jaki jest problem do rozwiązania?

Co nie funkcjonuje wystarczająco dobrze? Jakie element funkcjonowania instytucji lub osób prywatnych chcemy usprawnić? Dlaczego teraz nie jest ok?

Jaki jest cel projektu?

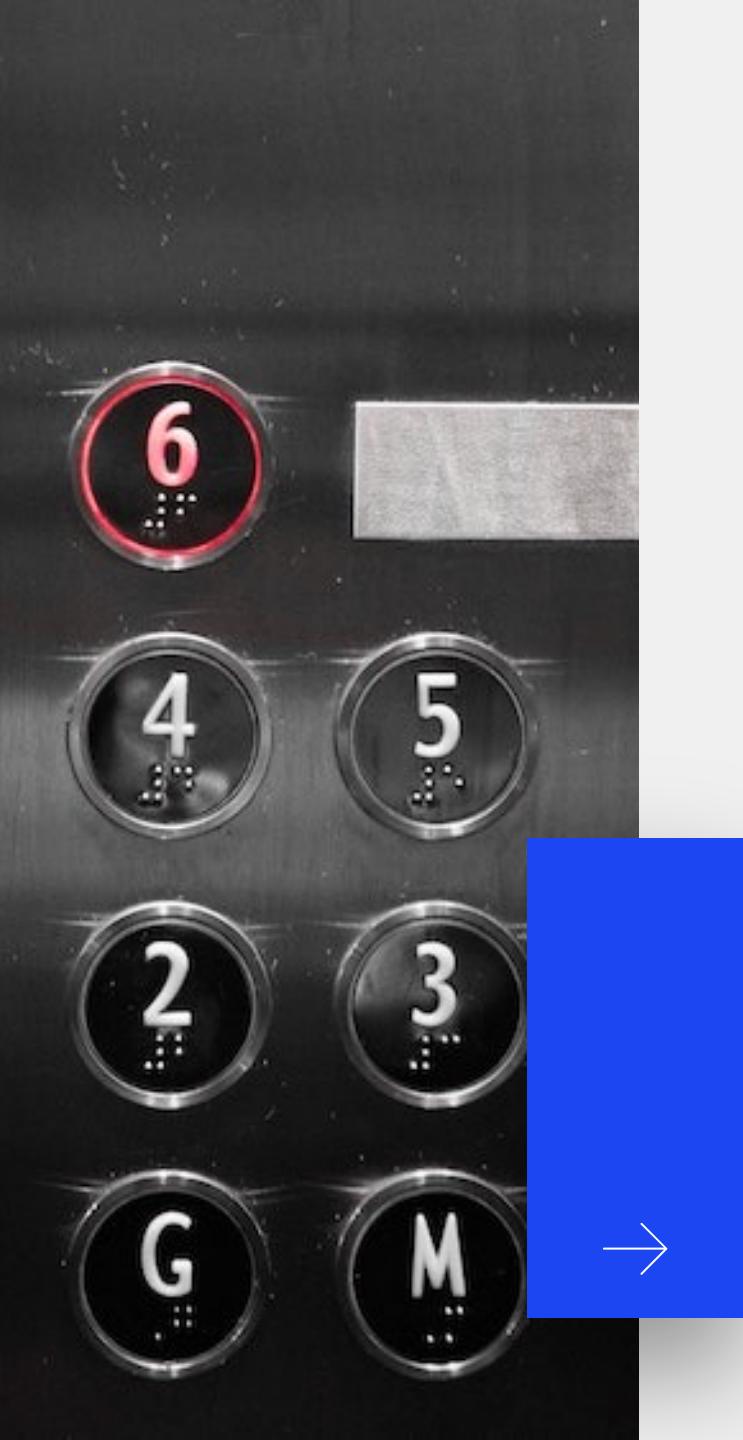
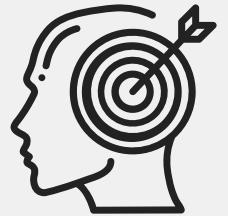
Motywacja do wykonania i wdrożenia produktu.

Jaki korzyści przyniesie nam wdrożenie produktu? Jakie skutki to powinno przynieść? -> Oczekiwania klienta.

Jakie są główne funkcje?

Jakie funkcje i dla kogo dostarczy planowany system?

Jaki będzie kształt systemu (aplikacja mobilna, webowa, złożony system)?



Wizja – Elevator Pitch

Dla <kto jest klientem?>, który <co potrzebuje?> <nazwa twojego produktu> jest <kategoria produktu>, pozwalającym na <główna zaleta>. W odróżnieniu od <nazwa głównego konkurenta>, nasz produkt jest <cecha odróżniająca>.

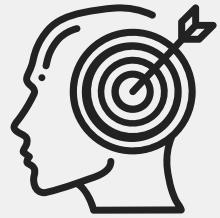
Elevator Pitch - przykład

Dla **<kto jest klientem?>**, który **<co potrzebuje?>** **<nazwa twojego produktu>** jest **<kategoria produktu>**, pozwalającym na **<główna zaleta>**. W odróżnieniu od **<nazwa głównego konkurenta>**, nasz produkt jest **<cecha odróżniająca>**.

—

“For a mid-sized company’s marketing and sales departments who need basic CRM functionality, the CRM-Innovator is a Web-based service that provides sales tracking, lead generation, and sales representative support features that improve customer relationships at critical touch points. Unlike other services or package software products, our product provides very capable services at a moderate cost.”

<https://www.joelonsoftware.com/2002/05/09/product-vision/>

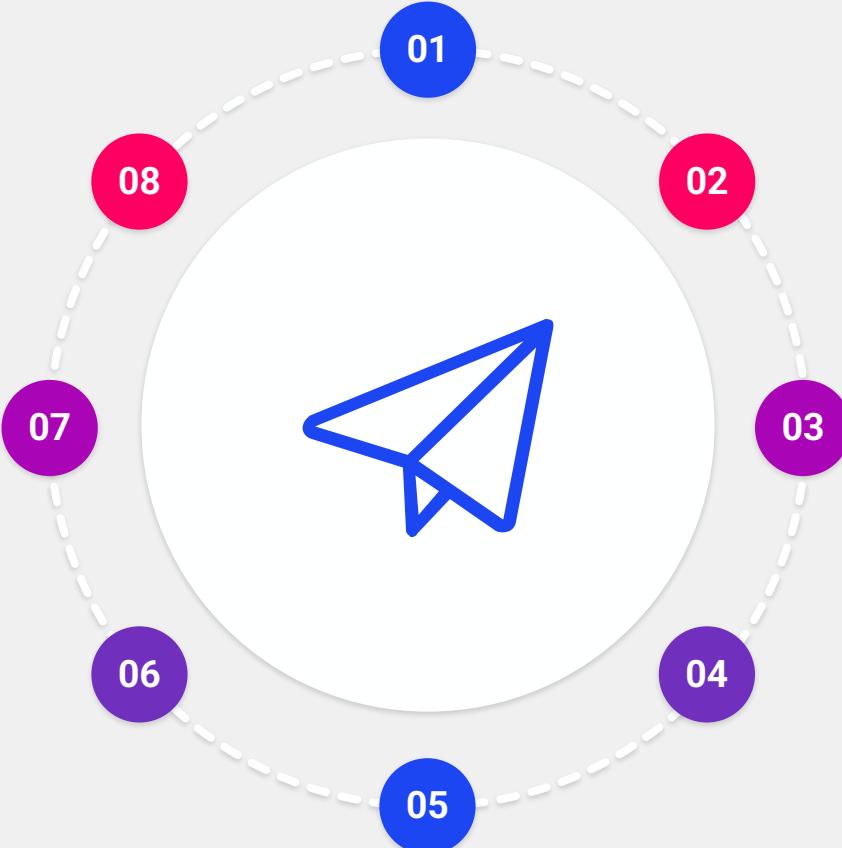


Dokument wizji

Dokument zawierający:

- **Opis dziedziny problemu** – nie zawsze potrzebne, ale bardzo przydatny w sytuacji gdy mówimy o złożonej dziedzinie, nieznanej dla zespołu realizującego produkt;
- **Charakterystykę problemu lub niedogodności**; aktualną sytuację, która przeszkadza w funkcjonowania organizacji / użytkownika; opis przyszłych użytkowników;
- **Cele projektu** – główne założenia dot. docelowego produktu, które rozwiążą zarysowany wcześniej problem; kształt systemu, motywacje jakie mają poszczególni użytkownicy
- **Porównanie z konkurencyjnymi / podobnymi** rozwiązaniami
- **Użytkowników / aktorów i systemy zewnętrzne**
- **Główne funkcje systemu** – na ogólnym poziomie
- **Główne założenia technologiczne** – założenia i istniejące ograniczenia

Pitch deck



Przyjrzyj się konkurencji!



Analiza konkurencyjnych rozwiązań jest często lekceważona, a tymczasem:

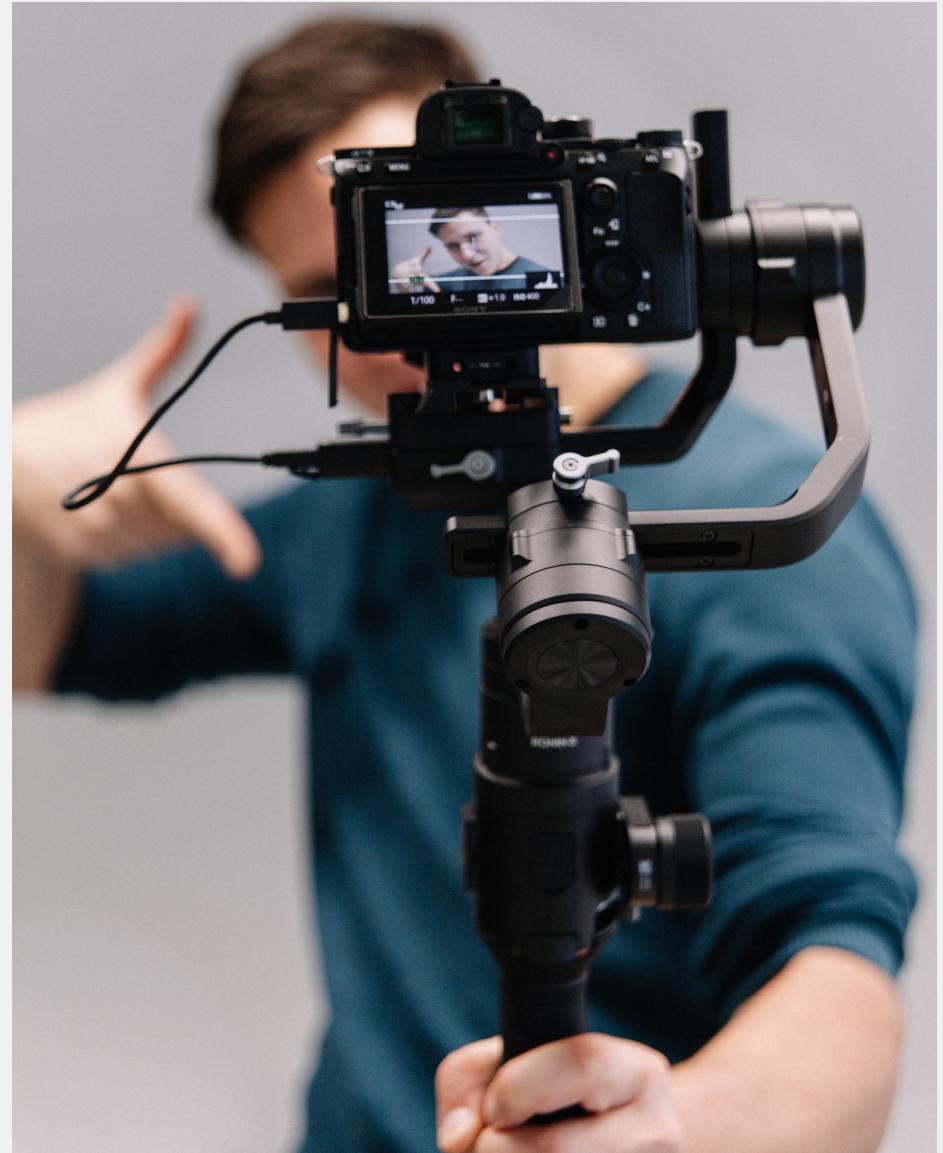
- Lepiej poznajemy dziedzinę problemu, potrzeby i oczekiwania klienta
- Podglądamy sprawdzone i chybione rozwiązania
- Może okazać się, że nie ma sensu realizować projektu
- Może okazać się, że można zrealizować projekt wykorzystując inne gotowe rozwiązanie lub na nim się (częściowo) wzorować

Zidentyfikuj aktorów systemu

Aktor:

- określa rolę odgrywaną przez użytkownika lub inny system (zewnętrzny), który wchodzi w interakcję z projektowanym systemem
- pojęcie zdefiniowane w standardzie UML

Aktor ilustruje osobną perspektywę użytkowania systemu.



= Monitorowanie jakości powietrza



Problem & rozwiązanie



Brak wystarczającej informacji

Mieszkańcy miasta mają słaby dostęp do informacji o jakości powietrza w swoim miejescu zamieszkania lub pracy



Nieefektywne kanały komunikacji z mieszkańcami

Obecne kanały komunikacji miasta z mieszkańcami są niewystarczające, nie umożliwiają np. efektywnego kontaktu z kierowcami



Brak możliwości badania zmian jakości

Miasto widzi bieżące stężenie poszczególnych zanieczyszczeń, ale nie ma mechanizmów do badania zmian w poszczególnych dzielnicach / obszarach



Problem & rozwiązanie



Gęsta infrastruktura czujników

Tanie czujniki pozwalają na umieszczenie ich w wielu miejscach, dzięki czemu możliwe będzie dokładne monitorowanie poszczególnych obszarów



Aplikacja mobilna z ostrzeżeniami

Aplikacja prezentująca aktualny stan jakości powietrza oraz notyfikująca użytkownika o złej jakości powietrza w danym miejscu oraz o komunikatach miasta



Aplikacja analityczna

Aplikacja przeznaczona dla Miasta, umożliwiająca badanie trendów oraz porównywania stanu jakości powietrza w różnych okresach czasu

Użytkownicy / aktorzy

Mieszkańcy miasta

- Podgląd biącego stanu jakości powietrza w pobliżu miejsca przebywania lub w inny miejscu wskazanym przez użytkownika
- Możliwość automatycznego odbioru powiadomień o złym stanie jakości powietrza
- Możliwość automatycznego odbioru ogłoszeń od UM

Urząd Miasta – urzędnicy odpowiedzialni za ochronę środowiska

- Pokazanie mapy jakości powietrza dla całego miasta oraz poszczególnych obszarów
- Możliwość wygenerowania zetwień porównujących jakość powietrza w poszczególnych miesiącach i latach
- Możliwość wysyłania wiadomości do mieszkańców, np. dzień darmowej komunikacji miejskiej, zakaz wjazdu pojazdów do danego obszaru / dzielnicy

Po co robić wizję produktu?

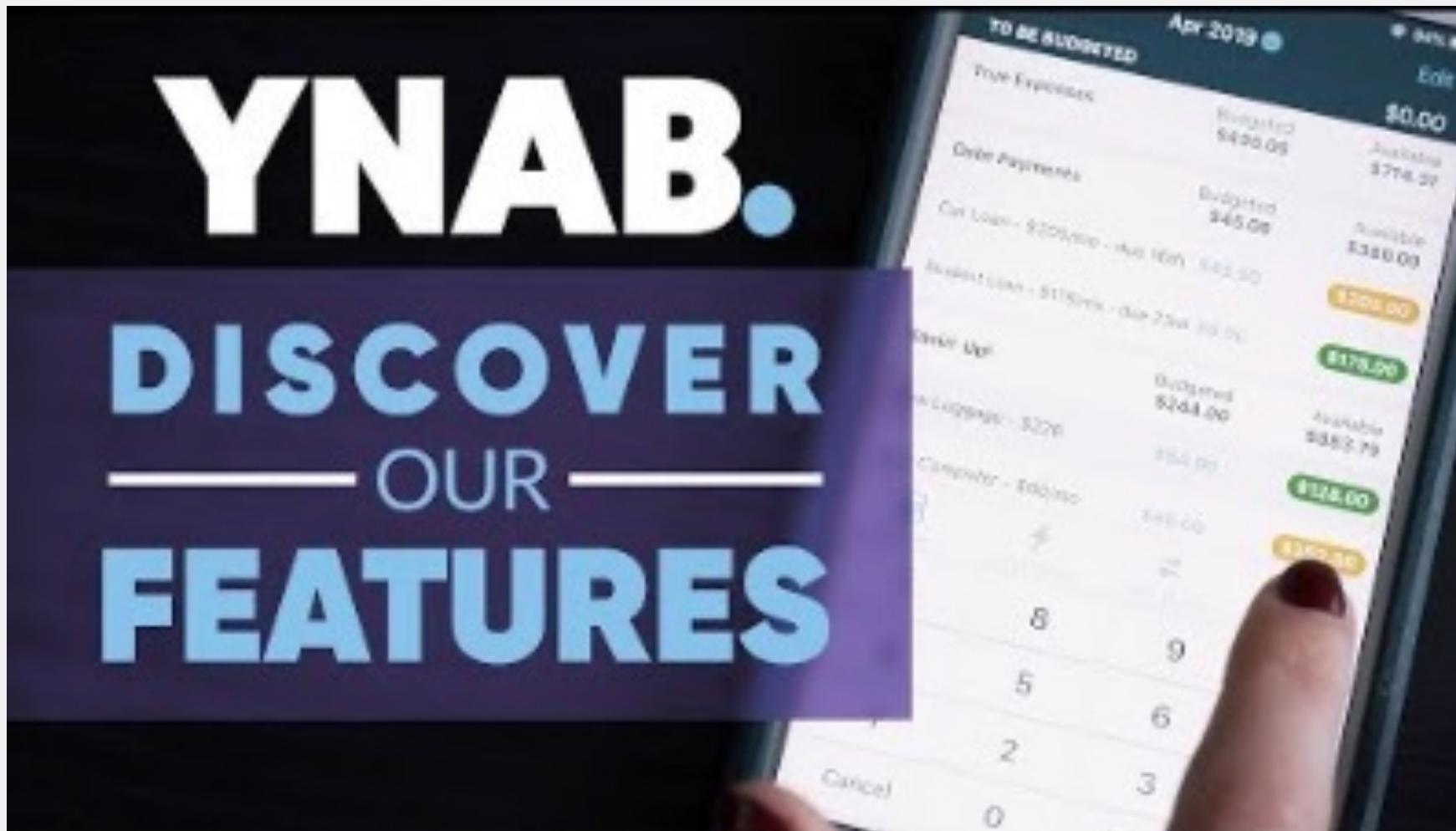
- Chcemy zrozumieć jakie problemy ma rozwiązywać produkt, co pozwoli nam pomóc w kształtowaniu produktu oraz upewnić się, że jesteśmy na dobrej drodze do sukcesu (i.e. klient będzie zadowolony)
- Pozwoli nam zidentyfikować odbiorców i dziedzinę, w której będzie działał system
- Chcemy zrozumieć co jest do zrobienia i zaplanować docelowy kształt procesu (na ogólnym poziomie abstrakcji)
- Dla klienta, aby zobaczył, że rozumiemy o co mu chodzi, czego potrzebuje. Jeśli będzie czuł, że go rozumiemy, to bardziej uwierzy w projekt i zaangażuje się w jego realizację.

=

Jak to robią inni?



Przykład



Przykład

The image shows a screenshot of a Slack message card. At the top, there is a dark purple header bar. Below it, the main card has a white background. On the left side of the card, there is a yellow square placeholder for a user profile picture, which contains a photo of a woman with her hand near her chin, looking thoughtful. To the right of this placeholder, the text "What is Slack?" is displayed in a large, bold, black font. At the bottom of the card, there is a dark purple footer bar. In the center of this footer bar is a white circular icon containing a yellow lightbulb. To the right of the lightbulb, the number "1" is displayed, indicating one unread message or notification.

slack

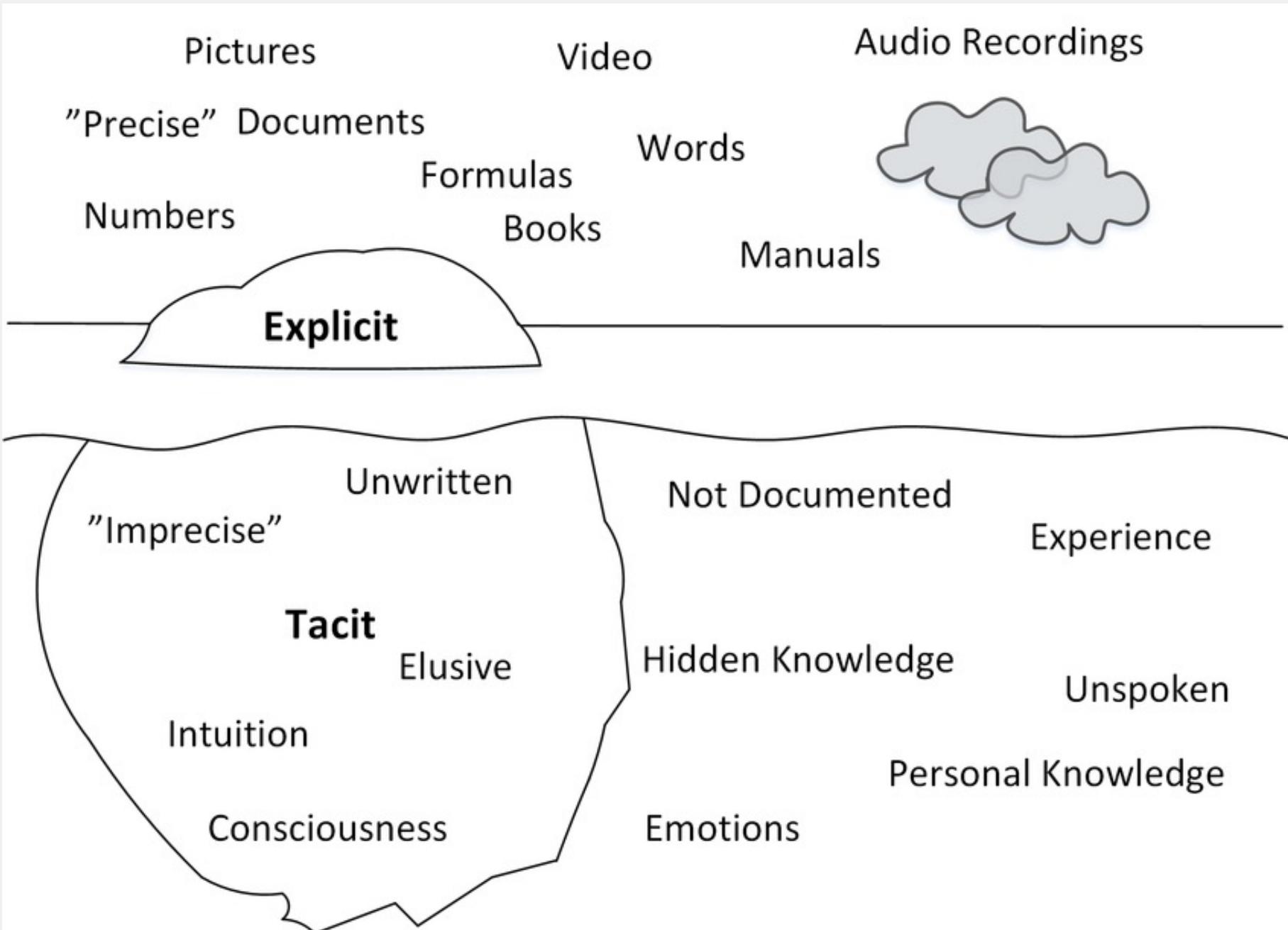
= Jak przygotować wizję?



Wiedza ukryta - Tacit knowledge

- Pojęcie, które przypisuje się Michael Polanyi
“możemy wiedzieć więcej, niż możemy powiedzieć”
- Wiedzę ukrytą przekazuje się w trakcie wspólnej pracy, rozmowy, opowiadania, wspólnych doświadczeń. Trudno ją wyartykułować, opisać, zawrzeć w specyfikacji
- Osoba posiadająca wiedzę ukrytą często nie jest jej świadoma.





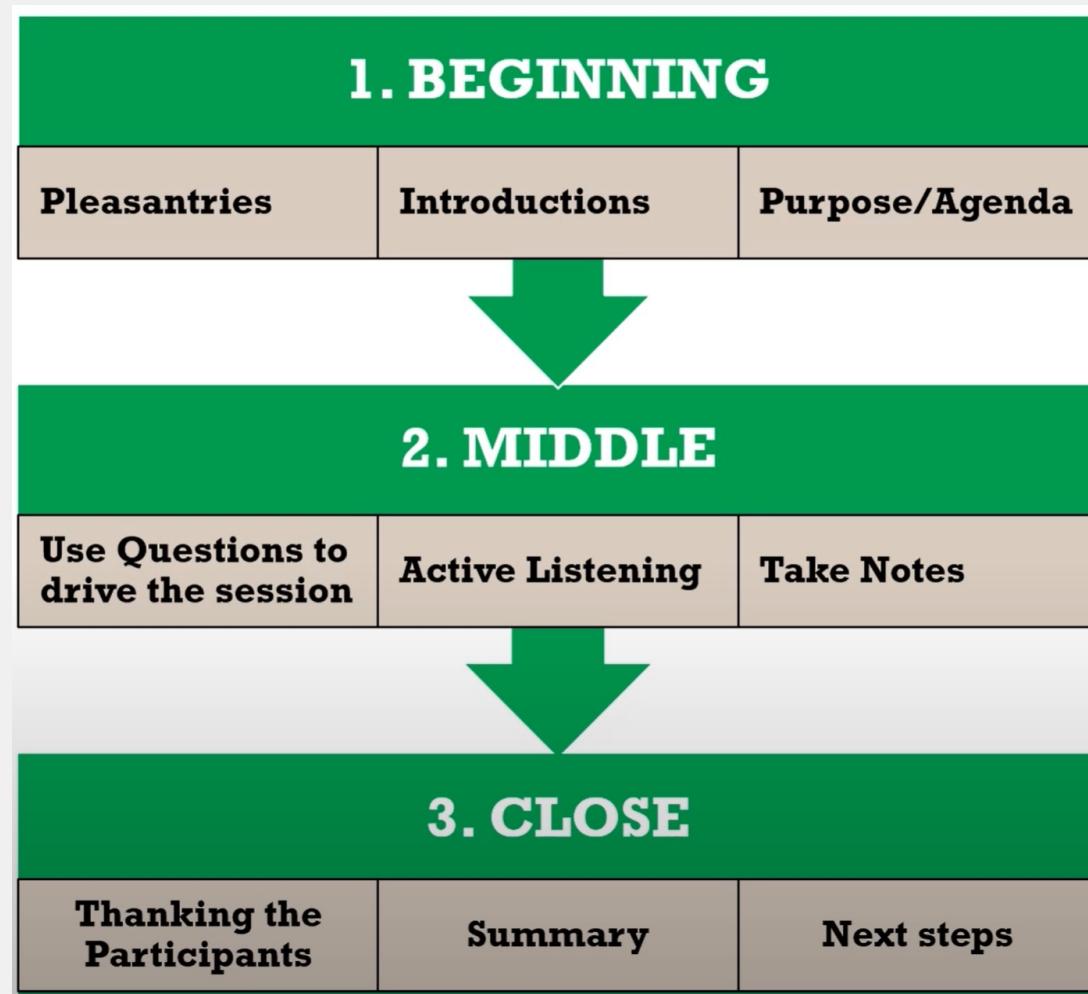
Pytanie otwarte vs. pytania zamknięte

- Potrzeby naszych przyszłych użytkowników mogą się znajdować w obszarze wiedzy ukrytej. W tym kontekście rozmowa z naszym klientem lub przyszłym użytkownikiem powinna być odpowiednio przeprowadzona
- Pytania powinny być typu otwartego żeby wydobyć wiedzę ukrytą klienta (pytania typu otwartego to pytania typu „w jaki sposób...”, „co się dzieje...”, „jak zachowuje się...”, „jak ma działać...”)
- Przykład: klient chce prezentacji, o co go zapytamy?
 - Czy dane mają być prezentowane w tabelce?
 - W jaki sposób mają być prezentowane dane?

Jak rozmawiać z klientem

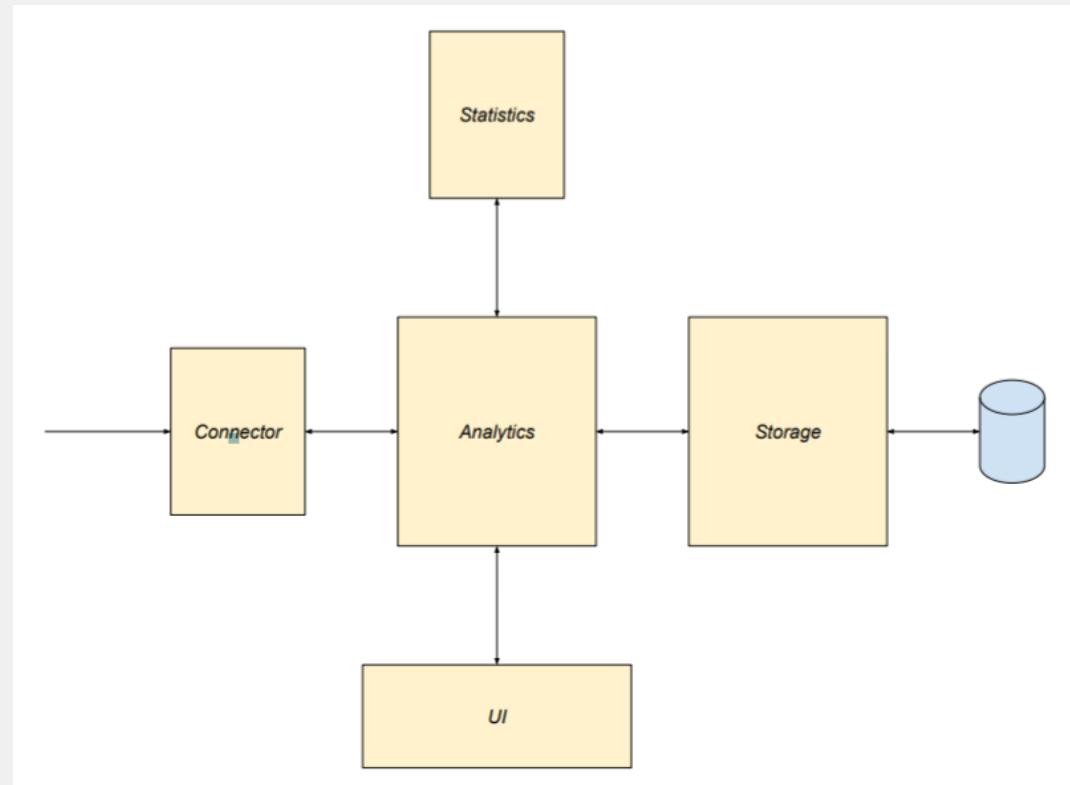
- Celem pierwszych rozmów jest eksploracja przestrzeni wymagań i identyfikacja potencjalnych problemów
- Nie wnikajmy w szczegóły chyba, że są kluczowe dla powodzenia projektu, na szczegóły będzie czas w kolejnych iteracjach analizy
- Szczegóły nie zaburzają rozumienia całości i celu projektu
- Pytania otwarta pomagają w eksploracji przestrzeni rozwiązań
- Pytania zamknięte pomagają w doprecyzowaniu wymagań

Jak rozmawiać z klientem

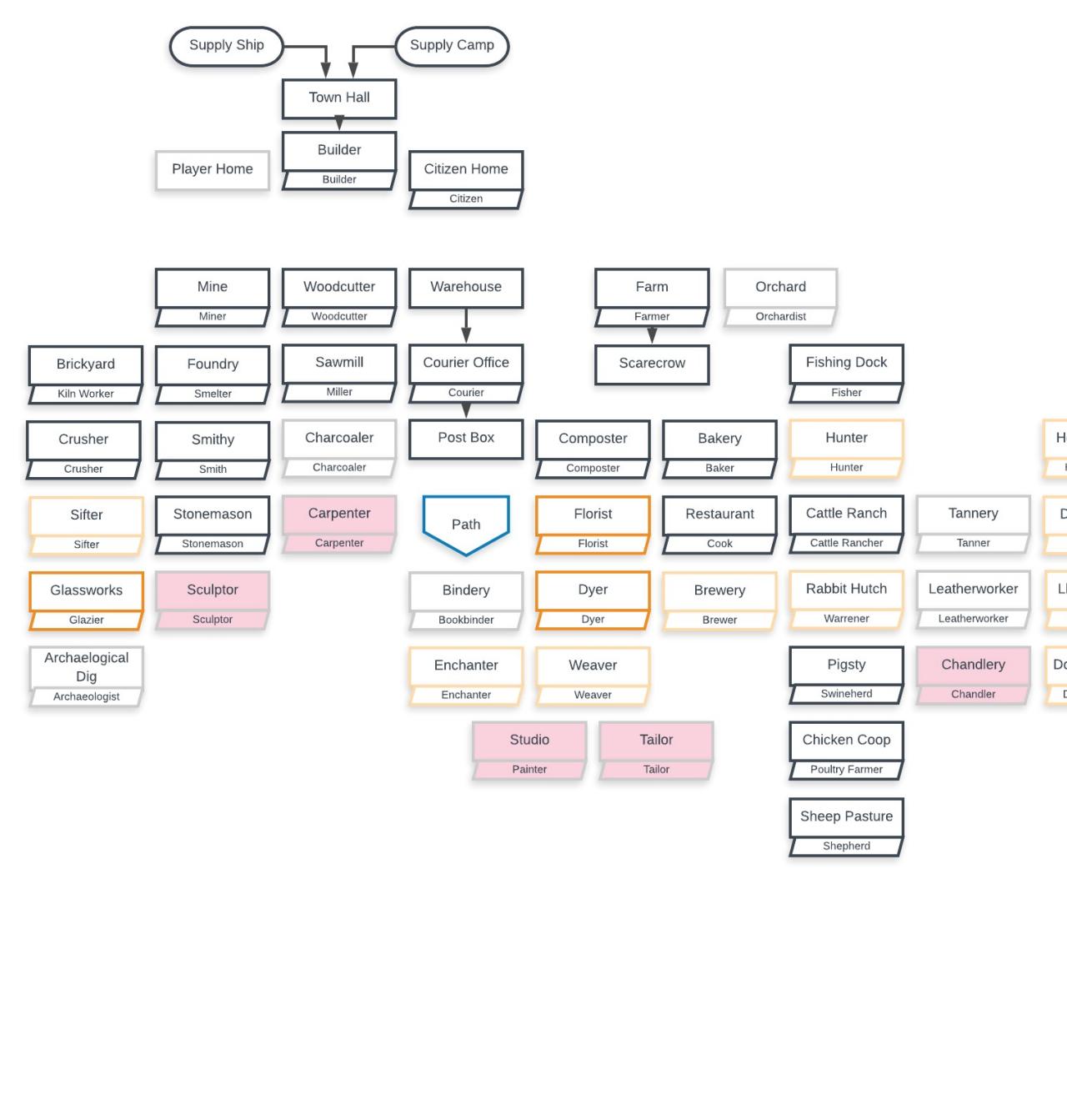
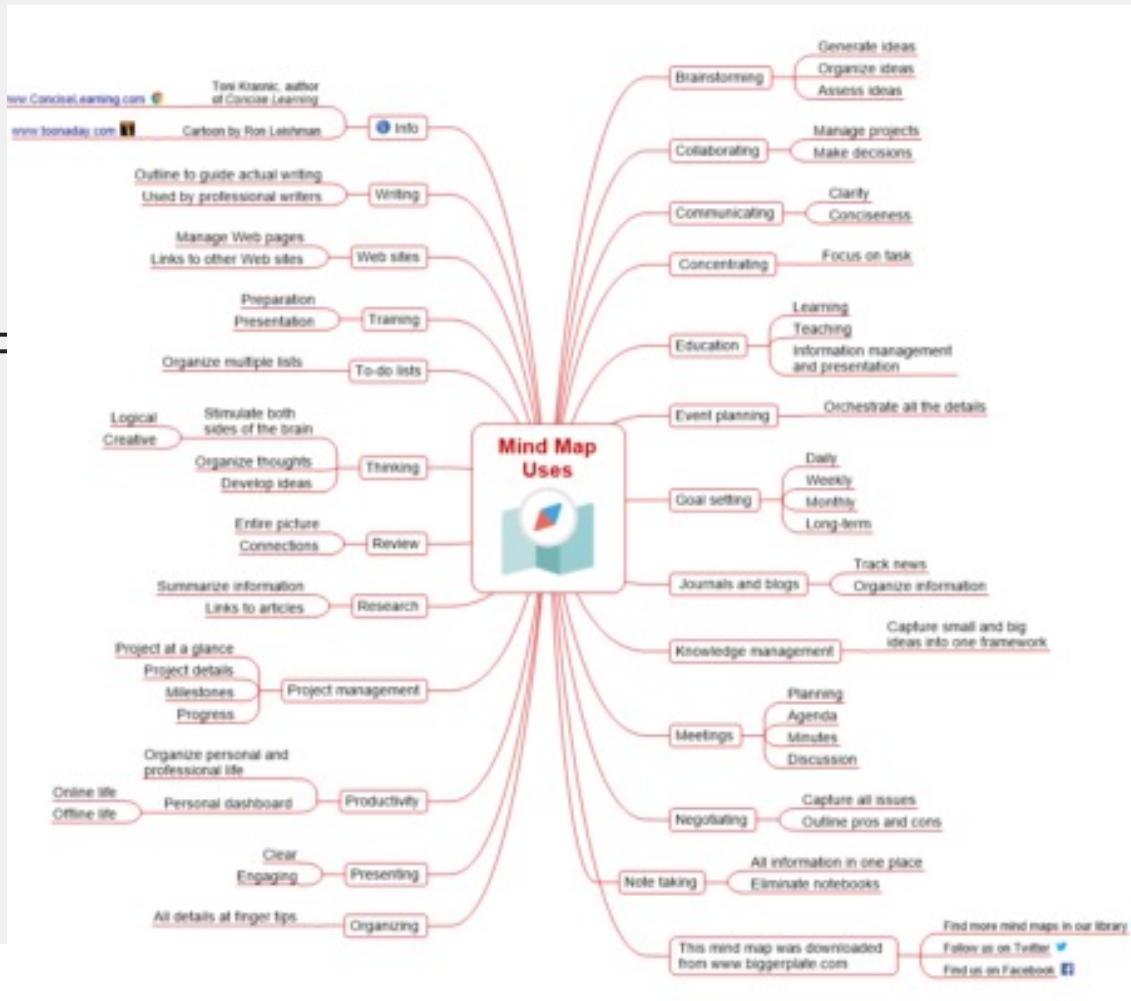


Przydatne techniki

- Lista funkcji
- Mind mapa
- Mapa funkcji
- Rysunek koncepcyjny



Przydatne techniki



Przydatne techniki

Item

1 REQ019 - Manage Inventory

The system MUST include a complete inventory management facility to store and track stock of books for the on-line bookstore.

1.1 REQ122 - Inventory Reports

Inventory reports are required that detail the available stock for each item including back orders. Future stock level reports should be able to predict the quantity of stock at a specified future date.

1.2 REQ023 - Store and Manage Books

A book storage and management facility will be required.

1.2.1 REQ022 - Order Books

A book order facility will be required to allow on-line ordering from major stockist's.

1.2.2 REQ021 - List Stock Levels

A facility will exist to list current stock levels and to manually update stock quantities if physical checking reveals inconsistencies.

= Jeśli nie zapamietam nic
innego to chociaż...



Ściąga



Rola inżynierii wymagań

- Zrozumienie realnych potrzeb użytkowników, identyfikacja problemów, które ma rozwiązywać system informatyczny
- Podstawa do dobrego projektu systemu oraz priorytetyzacji poszczególnych funkcji
- Medium w komunikacji pomiędzy klientem / użytkownikami a zespołem wykonawczym – produkt dopasowany do realnych potrzeb



Metodyki wytwarzania oprogramowania

- **Model kaskadowy** – specyfikacja wymagań wykonywania w pierwszym etapie (często potrzebna do określenia kosztów i czasu)
- **Modele iteracyjne** – proces specyfikacji wymagań rozłożony na etapy
- **Model adaptacyjny (Agile)** – zakłada zmienność wymagań przy określonym budżecie, czasie; doprecyzowanie wymagań w trakcie całego procesu

Ściąga

Wizja

Wizja to (powinien być) pierwszy krok w procesie wytwarzania oprogramowania, potrzebny do zdobycia inwestora, przekonania klienta lub określenia wspólnych celów wewnętrz zespołu wykonawczego.

Wizja powinna zawierać:

- Konkretny problem, który ma rozwiązywać nasz system
- Cele projektu pokazujące w jaki sposób problem zostanie rozwiązany za pomocą naszego systemu
- Do kogo skierowany jest produkt? Jakie będą mieć korzyści z wdrożenia / zakupu systemu?
- Główne funkcje produktu

Aktor

Użytkownik systemu lub zewnętrzny system (o tym będzie później), który korzysta z naszego oprogramowania z jakiegoś powodu – ma jasną motywację do użycia systemu. Aktor nie jest elementem systemu.



<https://www.menti.com/k8p1wptzo5> lub <https://www.menti.com> oraz kod: 4821 9380