

Budowa i integracja systemów informatycznych

dr hab. inż. Marta Łabuda
labudama@pjwstk.edu.pl

Laboratorium

Dr hab. inż. Marta Łabuda
Mgr inż. Grzegorz Cysewski
grzegorz.cysewski@pjwstk.edu.pl

Materiały wykładowe: M.Łabuda, St. Szejko



Materiały i zaliczenie

- **Udostępnienie materiałów**

<http://gakko.pjwstk.edu.pl>

- **Zaliczenie: teoria + projekt**

- **Ocena końcowa przedmiotu (wykładu)**

- **egzamin pisemny**

- **warunek dopuszczenia do egzaminu**

- : pozytywna ocena z projektu (oceny cząstkowe)

- : obecność na wykładach

- **warunek zdania**

- : pozytywna ocena z egzaminu pisemnego

- : ocena \Leftarrow $\frac{1}{2}$ sumy ocen za egzamin i za projekt, 50/50

Osoby, które nie zaliczyły ITN z BYT powinny powtarzać zajęcia projektowe

Treść przedmiotu BYT

Wykład stanowi omówienie podstawowych zagadnień inżynierii oprogramowania, w tym faz rozwoju oprogramowania oraz metod podwyższenia jego jakości; stanowi rozwinięcie wykładu „Projektowanie systemów informacyjnych” (PRI)

□ Wykład

- Motywacje i pojęcia IO; treść przedmiotu; Faza *Planowania projektu*
- Cykle życia oprogramowania - podstawowe modele; dobór strategii
- Procesy analizy i projektowania; wzorce, szablony i komponenty; integracja
- Zapewnianie jakości oprogramowania; bezpieczeństwo systemów
- Testowanie i walidacja oprogramowania; dokumentowanie testów.
- Wdrożenie i pielęgnacja oprogramowania

□ Projekt/laboratorium

Konstrukcja aplikacji w oparciu o przyjętą metodykę, wzorce (analityczne, projektowe) frameworki i komponenty + dokumentacja projektu

Literatura

Literatura podstawowa:

- Martin R. Czysta architektura. Struktura i design oprogramowania. Przewodnik dla profesjonalistów, Helion, 2022.
- Martin R. Czysty kod. Podręcznik dobrego programisty, Helion, 2023.
- Śmiałek M., Rybiński K. Inżynieria oprogramowania w praktyce. Od wymagań do kodu z językiem UML, Helion, 2024.
- Sommerville I. Inżynieria oprogramowania, WNT 2015.
- Górski J. (red.): Inżynieria oprogramowania w projekcie informatycznym, Wyd. II, 2000.
- Farley D. Nowoczesna inżynieria oprogramowania, Helion, 2023.
- Beck K. Uporządkowany kod. Ćwiczenia z empirycznego projektowania oprogramowania, Helion, 2024.

Literatura uzupełniająca:

- Khnonov V. Koncepcja Domain-Driven Design. Dostosowywanie architektury aplikacji do strategii biznesowej, Helion, 2022.
- Drąg J. Git. Od koncepcji do praktyki, Helion, 2023.
- Stellman A., Greene J. Agile. Przewodnik po zwinnych metodykach programowania, Helion, 2022.
- Smith B. DevOps dla zdesperowanych. Praktyczny poradnik przetrwania, Helion 2024.
- Żmigrodzki M. Zarządzanie projektami dla początkujących. Jak zmienić wyzwanie w proste zadanie. Wydanie III, Helion, 2021.

oraz wiele innych źródeł www dotyczących narzędzi i technologii IT

Literatura podawana także na bieżąco w treści wykładów

Skąd biorą się projekty IT?

➤ **Potrzeby / wyzwania rynku**

- np. ktoś zamierza stworzyć bank lub sklep internetowy, albo ogólnie – podnieść swoją konkurencyjność poprzez środki informatyki

➤ **Potrzeby biznesowe**

- plany strategiczne
- np. uświadomiona potrzeba z informatyzowania przyjmowania i przetwarzania zamówień składanych przez klientów firmy

➤ **Zamówienie przez Klienta**

- np. Kino zamawia aplikację do sprzedaży biletów przez Internet

➤ **Rozwój technologiczny**

- np. potrzeba przeniesienia danych do nowego SBD, upgrade'u oprogramowania, poprawa zabezpieczeń,...

➤ **Wymagania prawne**

- np. wymóg komputerowej certyfikacji jakiegoś produktu, albo zmiana przepisów podatkowych

*Niekiedy mogą prowadzić
do różnych projektów!!*

Sytuacja

Oprogramowanie nie jest takie, jakiego oczekuje klient !

Główne „osiągi” krytyczne:

- niepełna funkcjonalność, kiepska użyteczność, słabe dostosowanie do potrzeb
- za długo trwa przygotowanie oprogramowania
- za dużo kosztuje



© Scott Adams, Inc./Dist. by UFS, Inc.

Motywacja

Odpowiedzi na podstawowe pytania / problemy inżynierii oprogramowania, m.in.:

- Jak zbudować system / aplikację, jak dopasować do potrzeb, jaką wybrać drogę? Zespołowo?
- Aby było efektywnie, relatywnie tanio, a przede wszystkim – by system był dobrej jakości? I co to znaczy *dobrej jakości*?
- Jak zaplanować realizację systemu, co taki plan powinien zawierać?
- Jakiego winny być kolejne kroki procesu analizy wymagań? Projektowania? Testowania? Integracji komponentów?
- Co obejmuje *wdrożenie systemu*?
- Co z dokumentacją?

.....



Potrzeba skutecznych, efektywnych metod konstrukcji (...) oprogramowania wysokiej jakości

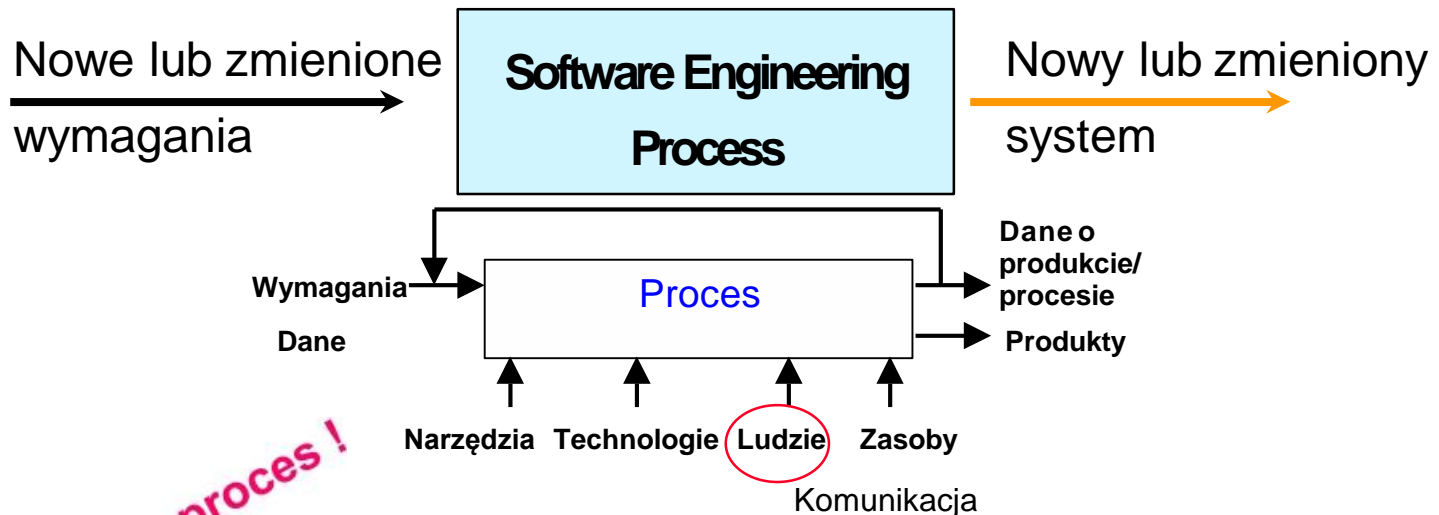
Remedium:

profesjonalizm – umiejętności,
kompetencja, etyka – w działaniach
inżynierii oprogramowania



Proces programowy

Proces programowy – każde działanie (zbiór działań) w ramach wytwarzania lub eksploatacji oprogramowania



Nie istnieje idealny proces !

- Zgodnie z przyjętym cyklem
- Adaptacja do potrzeb organizacji, projektów, technologii, ludzi
- Proces określa **jakie działania**, **kto** wykonuje, **kiedy** i **jak**, aby osiągnąć założony cel

Proces programowy

Proces - sekwencja kroków podejmowanych w zadanym celu.

Proces programowy – zbiór działań, metod, procedur i przekształceń dokonywanych w celu budowy lub utrzymania oprogramowania oraz związanych z nim produktów.

- procesy podstawowe (wytwarzania, użytkowania, pielęgnacji,...)
- procesy organizacyjne (zarządzania, doskonalenia, ...)
- procesy wsparcia (weryfikacji, dokumentowania, zapewniania jakości,...)

Czy proces programowy = cykl życia oprogramowania? Tak, jeśli patrzeć na najwyższym poziomie; ale procesami są też różne składniki cyklu!!

Proces wytwórczy oprogramowania – proces, w którym potrzeby użytkownika są przekształcane w produkt programowy.

Projektem nazywane jest trwające pewien czas (tj. posiadające swój punkt startu i punkt zakończenia) przedsięwzięcie, mające na celu wytworzenie pewnego produktu lub usługi.

Projekt informatyczny to całokształt planowych działań, technicznych, organizacyjnych i menedżerskich, mających na celu wytworzenie systemu, produktu lub usługi w dziedzinie informatyki; realizowany warunkach ograniczeń



Od problemu do realizacji

❑ IO podsuwa wzorce metodyk budowy oprogramowania

Tradycyjne...

- Kaskadowy (klasyczny)
- Model V
- Model prototypowy
- Model przyrostowy
- Model spiralny

Ich hybrydy i adaptacje



www.cagesbydesign.com

Podejście (paradygmat) – przyjęty sposób widzenia rzeczywistości w danej dziedzinie

Od problemu do realizacji

... i nowsze - **bardziej lub mniej udane** -

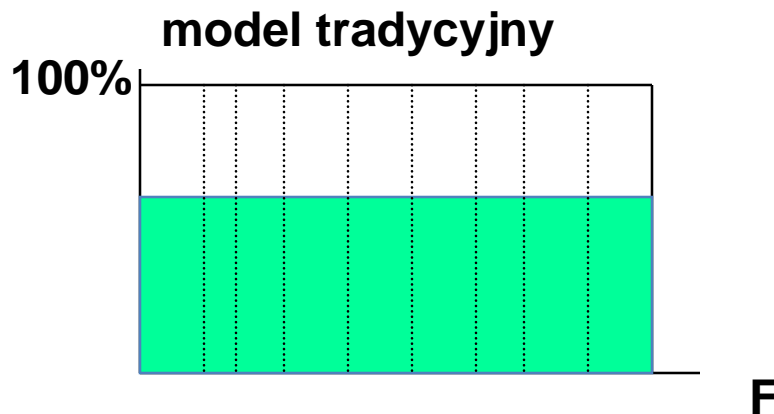
- Wielokrotne użycie oprogramowania (*ang. software reuse*)
- Model komponentowy
- Adaptacyjne (lekkie, *agile*) – SCRUM, *extreme programming*, *test-driven programming*, *design driven development*, *behaviour driven development*
- Przekształcenia modeli – MDD, DDD, MDA i przekształcenia formalne
- Ponowna inżynieria (*ang. re-engineering*)

Dlaczego? Skąd taka potrzeba?

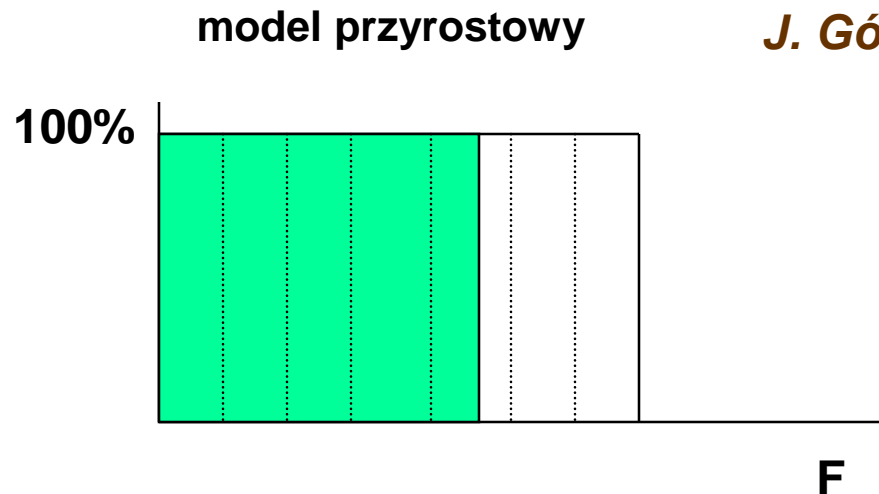


Przyrostowa budowa systemu

J. Górski



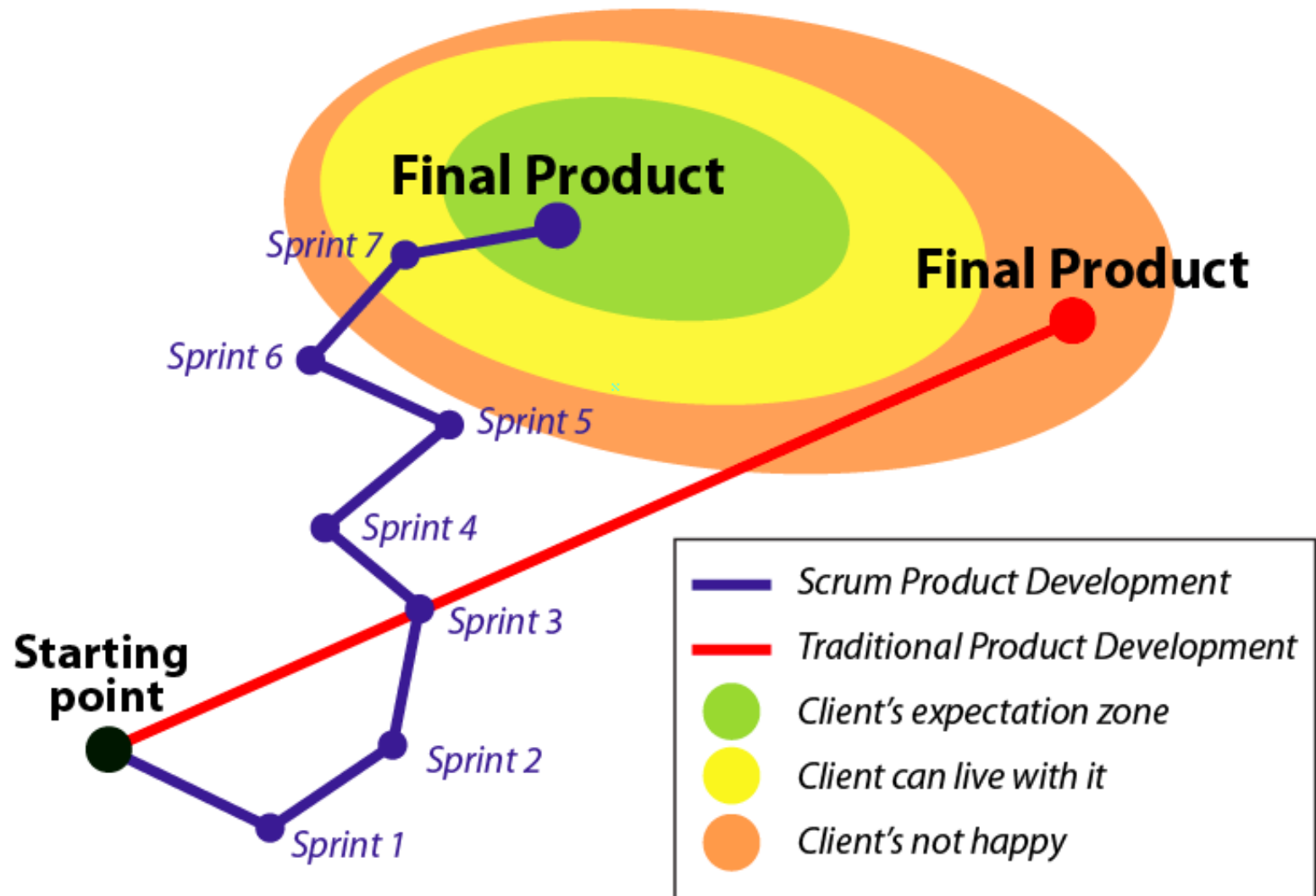
**W chwili T, 100% systemu
jest ukończone w 60%**



**W chwili T, 60% systemu
jest ukończone w 100%**

F - funkcjonalność systemu
T - czas

Scrum vs. Traditional



Od problemu do realizacji

- ❑ IO oferuje gotowe
- ✓ Komponenty
- ✓ wzorce
- konstrukcji
- oprogramowania
- ✓ szablony
- (*frameworki*)

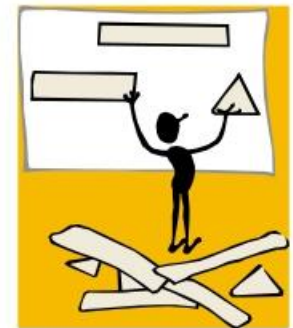


Jak poprowadzić przedsiębiorstwo, jak dobrać strategię postępowania?

Nie ma „jednej prawdy”, są tylko lepsze lub gorsze rozwiązania konkretnych sytuacji

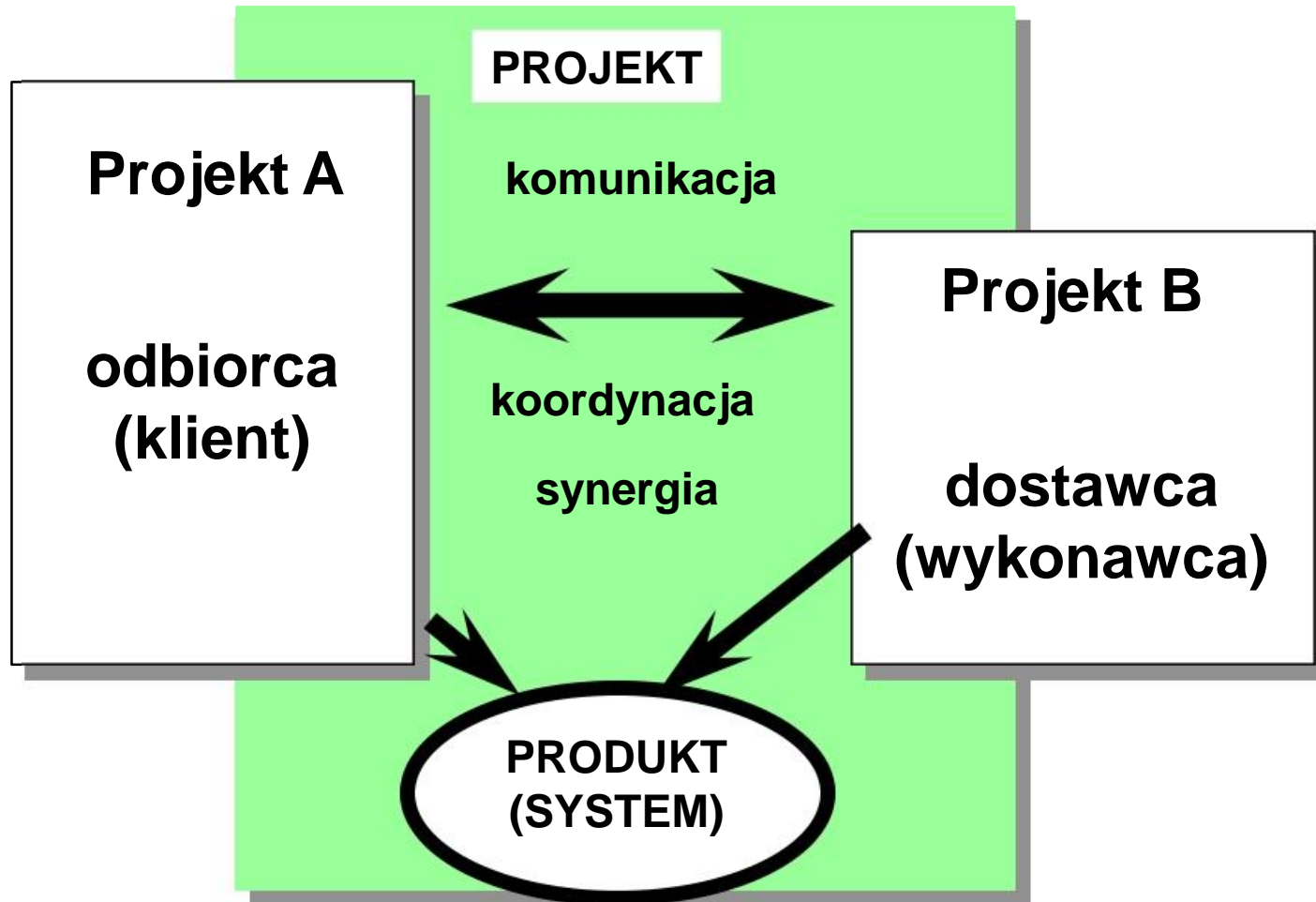
Czynniki wpływające na dobór strategii

- sytuacja problemowa
- modele, ich przydatność w tej sytuacji
- minimalizacja ryzyka
- sytuacja firm
- technologia
-



W praktyce modele (wzorce) rzadko występują 'w stanie czystym' – są dostosowywane do potrzeb / sytuacji projektów lub firm je realizujących

Przedsięwzięcie informatyczne to są dwa projekty! *J. Górski*



Procesy Klienta i Dostawcy

Faza	Projekt A (Kupującego)	Projekt B (Dostawcy)
Planowanie	Identyfikacja potrzeb i wymagań biznesowych Analiza wykonalności Podejmowanie decyzji Wstępne planowanie projektu Przydział zasobów	Lobbing Ewentualna analiza wykonalności - na zamówienie Kupującego
Negocjowanie	Organizacja procesu selekcji Kontraktacja	Ocena zasobów Ocena korzyści Podejmowanie decyzji Planowanie Udział w przetargu i negocjacje kontraktu
Wytworzenie	Uszczegółowienie wymagań Nadzór Walidacja i testowanie akceptacyjne	Analiza wymagań Projektowanie Kodowanie i testowanie Weryfikacja, walidacja i testowanie
Wdrożenie (Rozpowszechnienie)	Absorpcja systemu Eksploatacja	Procesy utrzymaniowe i ewolucji oprogramowania

Projekt informatyczny

Projekt informatyczny:

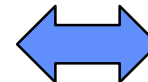
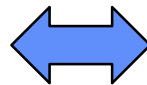
planowa działalność, obejmująca zestaw zadań mających wspólny cel – wytworzenie systemu informatycznego lub produktu (usługi) programowego; w warunkach ograniczeń

każda!

Sukces projektu:

*osiągnięcie zakładanego celu w zakładanym czasie
i przy zakładanych kosztach
+ satysfakcja klienta,
+ pozytywne skutki dla społeczności, jednostek*

- **systematyczna realizacja**
- **metodyczna**
- **udokumentowanie**
- **znajomość wszystkich aspektów**

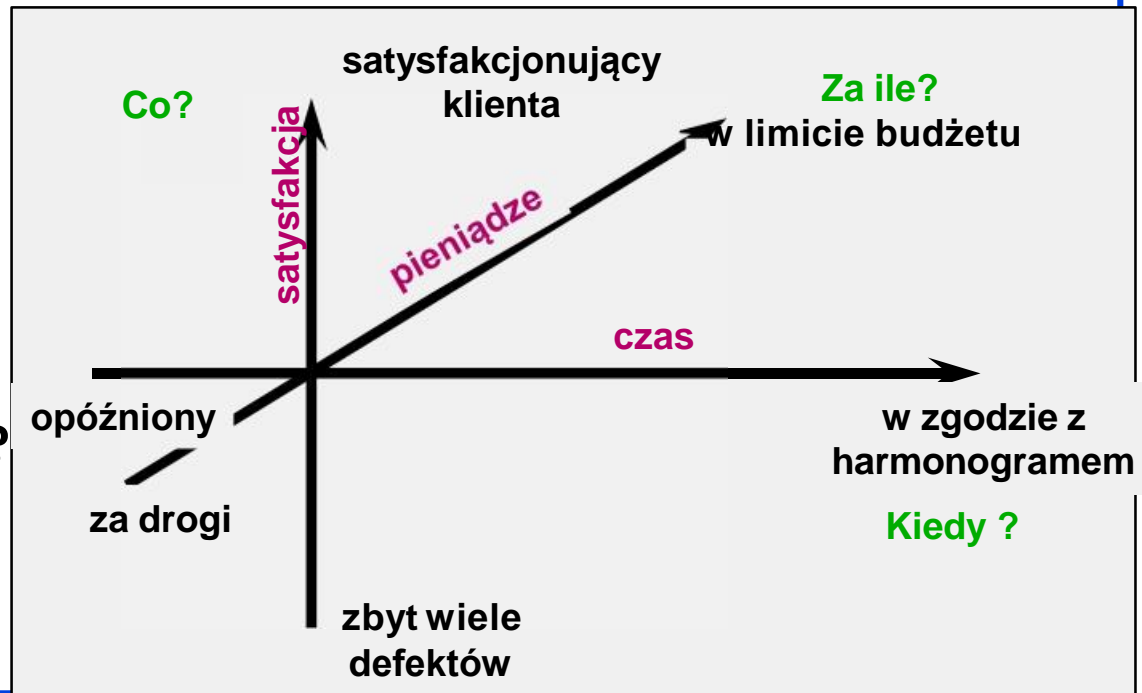


Klient

Miary sukcesu

Projekt osiąga swoje cele biznesowe interesariuszy w wyznaczonym czasie i w ramach przewidzianego budżetu

- cele (co chcemy osiągnąć?)
 - satysfakcja klienta
- harmonogram (w jakim czasie?)
- budżet (ile jesteśmy skłonni zapłacić?)
- wpływ na udziałowców? (skutki społeczne)



Właściwości projektu (zapewne każdego)

- **jednostkowość i złożoność**
- **ograniczoność środków**
- **wymaga pogodzenia pracy ludzkiej, zasobów, kapitału i czasu**
- **włącza w swój obręb różnych ludzi, funkcje i organizacje**
- **wymaga planowania, koordynacji i sterowania**

Właściwości projektu informatycznego

- konsekwencje *specyfiki produktu programowego*
- tendencja do jak najwcześniejszego podjęcia implementacji
- rozległość dziedzin problemów wymagająca specjalistycznej wiedzy
- niestabilność wymagań
- realizacja w układzie klient-dostawca

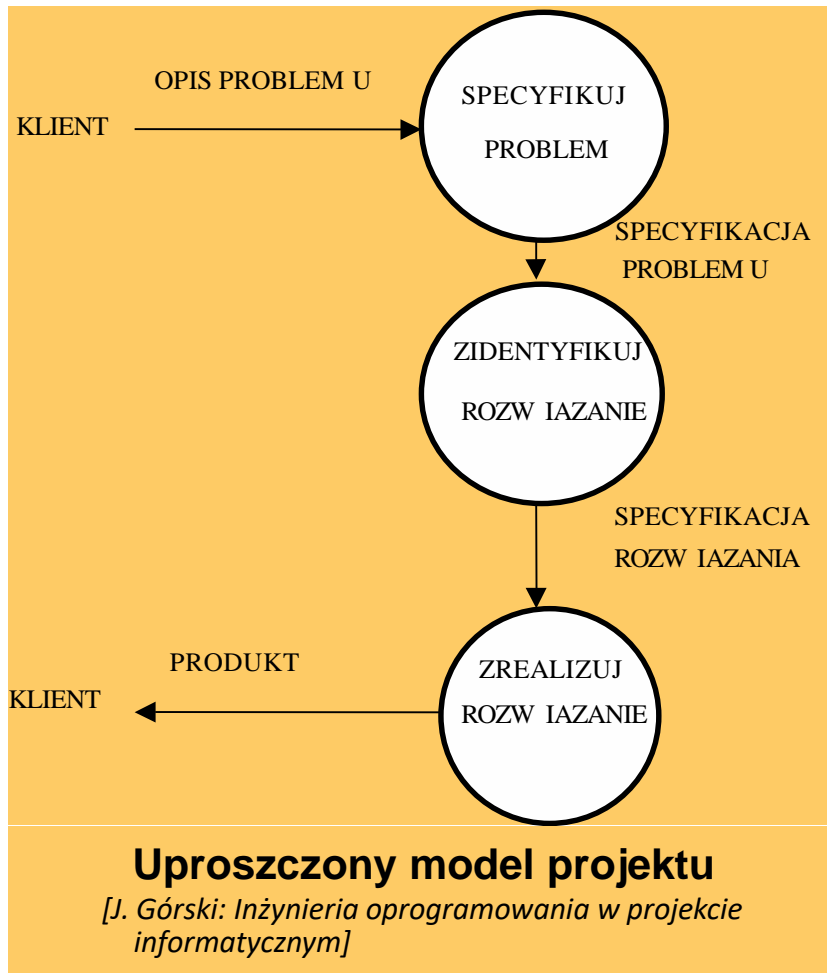
- trójkąt wymiarów projektu



Wybrane charakterystyki projektów informatycznych

- **Dziedzina**
- **Wielkość, złożoność projektu**
- **Innowacyjność / typowość projektu**
- **Czas na wykonanie projektu**
- **Znajomość i stopień stabilności wymagań**
- **Wielkość zespołu i umiejętności
poszczególnych osób**
- **Dalszy rozwój systemu**
- **...**

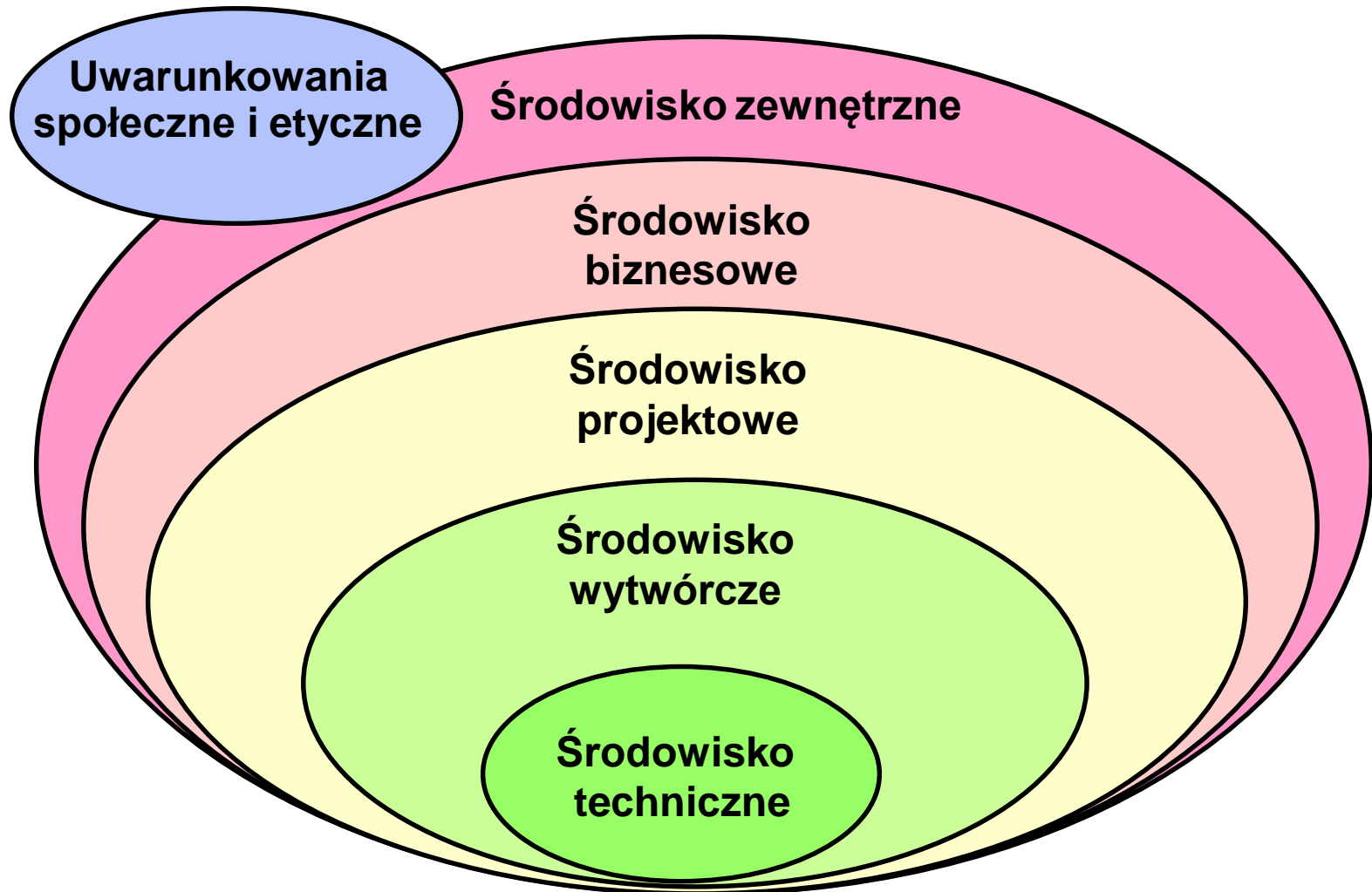
Model i ważniejsze atrybuty projektu informatycznego



- **początek, koniec**
 - *rola Klienta!*
- **fazy / etapy**
- **produkty**
 - etapowe, finalne
- **zadania**
 - techniczne i menedżerskie
 - zwykle dzielone na podzadania i czynności
- **czas trwania, koszt**
- **zasoby**
 - ludzie, infrastruktura,
 - czas
 - pieniądze, ...

Projekt w szerszym kontekście

J. Górski



Cele projektów

➤ Cele dalsze

- związane z korzyściami jakie chcemy uzyskać poprzez informatyzację (użycie / poprawę środków informatyki), np

Klient: zwiększenie wydajności produkcji, jakości obsługi, kompletności informacji albo zasięgu klientów naszego biznesu

Zamawiający: zdobycie renomy, realizacja kontraktu

➤ Cele bliższe

- Określające jaki produkt lub usługę należy dostarczyć poprzez przeprowadzenie projektu

np. analiza celowości..., aplikacja internetowa (z bazą danych), system sterowania, komunikacja pomiędzy stacjami nad/odb.,...

Osiągnięcie celów bliższych prowadzi do realizacji celów dalszych

Główne obszary prowadzenia projektu informatycznego (*zarządzania projektem*)

- ➡ koordynowanie prac, reagowanie na problemy i zmiany
- ➡ zarządzanie czasem i zakresem projektu
- ➡ zarządzanie kosztami
- ➡ zarządzanie infrastrukturą
środowiskiem, ludźmi, komunikacją,
wersjami, dokumentowaniem,
- ➡ zapewnienie jakości
- ➡ identyfikacja i przeciwdziałanie
zagrożeniom (ryzyku)
- ➡ zarządzanie konfiguracją

*Przewodnik Komitetu
Standaryzacyjnego PMI
(Project Management
Institute)*



IEEE Std. 1490-98

Interesariusze projektu

Interesariuszem projektu jest każdy podmiot (niekoniecznie ożywiony), który ma uzasadnione prawo wywarcia (pośrednio lub bezpośrednio) wpływu lub może znaleźć się pod wpływem rozpatrywanego projektu

Punkt widzenia - perspektywa widzenia projektu lub systemu; koncentracja uwagi na wybranym aspekcie



- **klient**
 - zamawiający, użytkownik, operator, ...
- **wykonawca, dostawca**
 - zespół projektowy
 - wykonawcy: analitycy, programiści, testerzy,...
 - poddostawcy, eksperci...
- **oprogramowanie, urządzenie współpracujące**
- **interesariusze pośredni**
 - klienci klienta, firma i klienci wykonawcy, zewnętrzni beneficjenci lub ofiary systemu, ...

Zagrożenie obniżenia poziomu sukcesu przedsięwzięcia = *ryzyko*



Co to jest „ryzyko”?

- możliwość zaistnienia niechcianego zdarzenia
- niepożądane konsekwencje



Zagrożenia w projekcie informatycznym - przykłady [J. Górski]

Dla klienta i wykonawcy

- przekroczenie budżetu
- przekroczenie terminu realizacji

Dla użytkownika końcowego

- niewłaściwa funkcjonalność
- “trudny” interfejs użytkowy
- niska wydajność systemu
- wysoka zawodność systemu

Dla wdrożeniowca

- niska jakość oprogramowania
- trudność w dopasowaniu systemu do środowiska docelowego

Budowa i integracja SI

Co zrobić z ryzykiem?

pasywna akceptacja ⇔ plan aktywnego przeciwdziałania



folia 43

J. Miller, WETI PG

□ Wizja systemu

cele, wymagania, rozwiązania projektowe

✓ **Cele** wyjaśniają **dlaczego potrzebujemy** systemu w kontekście organizacji

- jakie są spodziewane korzyści
- utracone korzyści gdy go nie będzie
- ... (**cele dalsze**)

- Poprawa obsługi klienta (nowe usługi)
- Zwiększenie konkurencyjności własnych produktów, poprawa jakości
- Lepsze szacowanie własnych zasobów
- Dodatkowe argumenty marketingowe
- Obniżka kosztów obsługi
- Zwiększenie własnej płynności finansowej
- ...

oczekiwane produkty i usługi (**cele bliższe**)



✓ **Rozwiązania** projektowe określają

- **zakres systemu** i wykorzystanie dla realizacji wymagań
- budowę i działanie systemu
- drogę dojścia do systemu docelowego

Wymagania przekazują uzasadnienie dla potrzeby wytworzenia systemu - określają pośrednio cele systemu i wyjaśniają na ile i dlaczego są ważne

✓ **Wymagania** względem systemu definiują:

- **usługi** dostarczane przez system jego otoczeniu,
- wymaganą **jakość** tych usług,
- określają **ograniczenia**, w ramach których system jest realizowany oraz użytkowany

Zakres projektu

Zakres projektu - zakres pracy koniecznej, by dostarczane oprogramowanie i usługi (**produkty projektu**) posiadały wymagane cechy

- **Zakres projektu** — **zadania i ich produkty**
- **Zakres projektu** — **pokrycie cyklu życia**
- **Zakres projektu a zakres systemu**



Cykl życia projektu

Cykl życia projektu – zbiór (zwykle sekwencja) faz projektu, od jego rozpoczęcia do zakończenia, wyodrębnianych z uwagi na potrzeby sterowania nim przez organizację (organizacje) zaangażowaną w projekt.

- **faza projektu** grupuje pewne zadania
- z fazą projektu związany jest zbiór jej *produktów*
- zwykle kończy się *przeglądem końcowym fazy*, robionym w celu
 - stwierdzenia wykonania jej produktów, wykrycia i usunięcia usterek
 - zdecydowania, czy projekt powinien przejść do fazy następnej

Cykl życia projektu określa

- ⇒ początek i koniec projektu
 - akcje związane z rozpoczęciem (*ustanowienie*) i zakończeniem projektu
- ⇒ wyodrębniane fazy, ich powiązanie, przekazywane produkty
- ⇒ zakres prac poszczególnych faz
 - często związany z technologią
- ⇒ wykonawców faz
- ⇒ sposób zarządzania projektem

***Nie należy utożsamiać cyklu życia projektu i
cyklu życia oprogramowania
(produktów programowych)!***

**Ale model wytwarzania lub
ewolucji silnie wpływa na cykl
(kształt, strategię prowadzenia)
projektu wytwórczego, ewolucji
oprogramowania**





POLSKO-JAPONSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Inżynieria oprogramowania

Obejmuje praktyczną stronę informatyki

obejmuje wszelkie aspekty produkcji oprogramowania: analizę i określenie wymagań, projektowanie, implementację, wdrożenie, pielęgnację i ewolucję gotowego produktu.



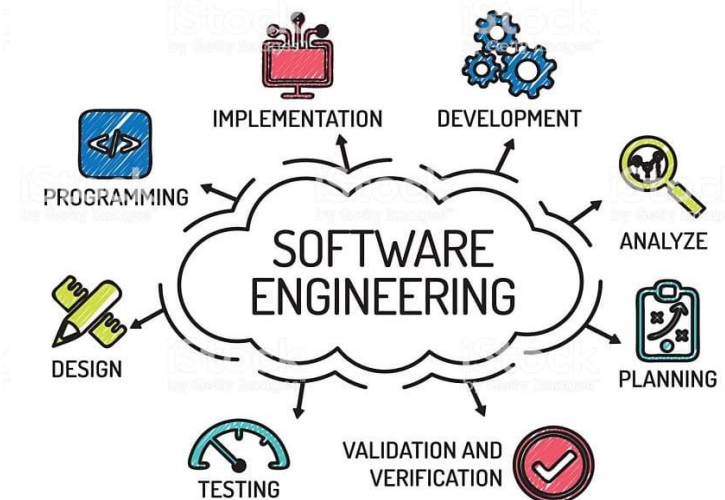
Obszary inżynierii oprogramowania

- Cykle życia (i wytwarzania) oprogramowania; ewolucja oprogramowania
- Podejścia i metodyki, np. obiektowość (UML)
 - metody modelowania, analizy, projektowania, testowania,...
- Inżynieria wymagań
- Zagadnienia profesjonalizmu
 - stosowanie metodyk, standardów, narzędzi; aspekty prawne, etyczne
- Miary i oceny jakości oprogramowania
- Środowiska i narzędzia wspomagające, w tym narzędzia CASE
- Proces wytwórczy
 - kształt, zarządzanie, ocena, poprawa, ryzyko,...
- Języki i techniki programowania



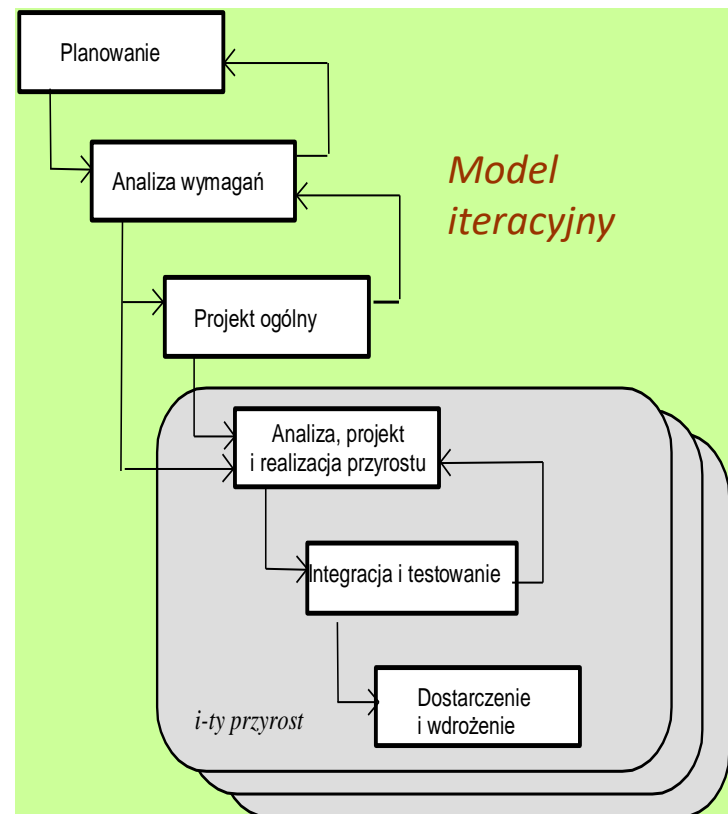
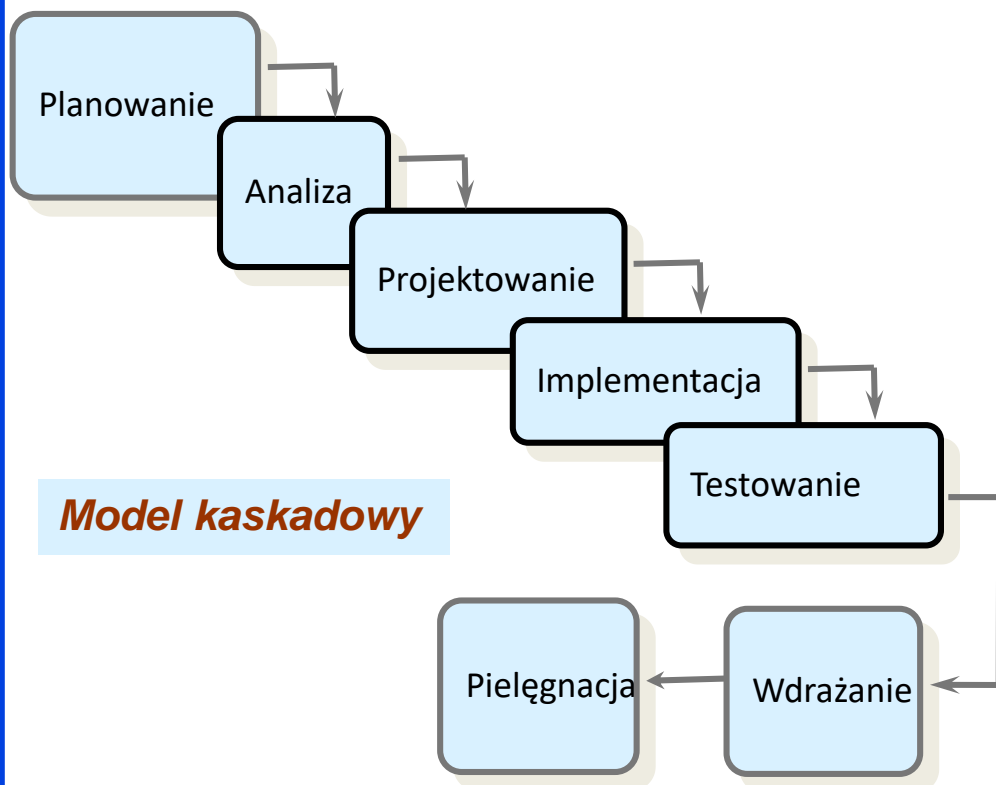
Znaczenie inżynierii oprogramowania

- gospodarki wszystkich rozwiniętych krajów zależą od oprogramowania
- ponadto, samo wytwarzanie oprogramowania jest też poważną gałęzią gospodarki narodowej każdego rozwiniętego kraju
- coraz więcej i więcej systemów wymaga niezawodnego oprogramowania
- istnieje potrzeba systematyzacji i uporządkowania procesu wytwarzania oprogramowania celem ułatwienia tworzenia oprogramowania wysokiej jakości

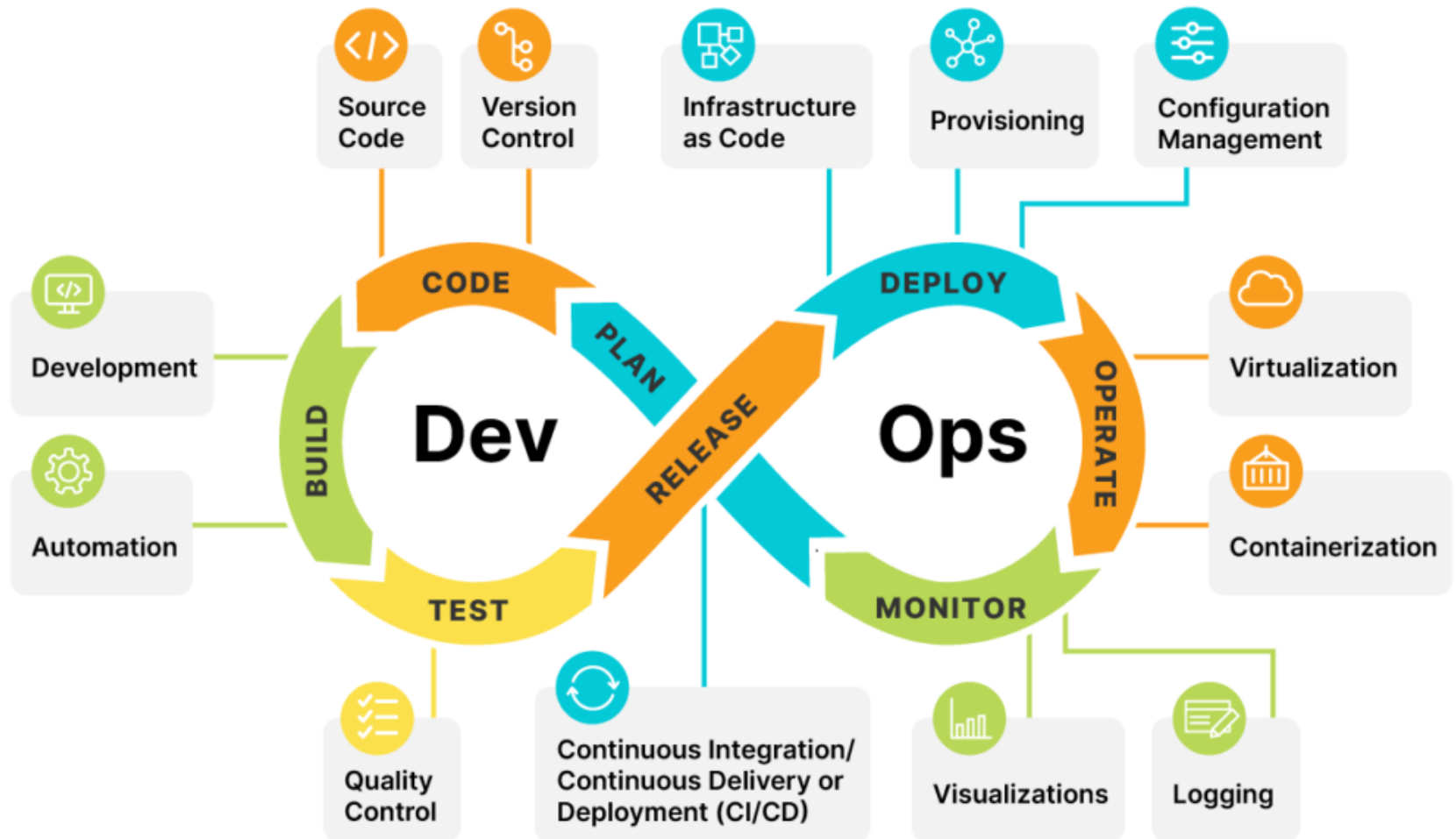


Wytwarzanie i pielęgnacja systemów

IO ujmuje wytwarzanie i eksploatację oprogramowania w pewne etapy, definiując różne **modele** (wzorce) **procesu twórczego** i **utrzymaniowego** i dostosowując metodyki twórcze do tych modeli



Wytwarzanie i pielęgnacja systemów



Inżynier oprogramowania (software engineer)

- zajęcie wpisane na oficjalną listę zawodów w roku 1990 (za granicą)
- według *Money Magazine* i *Salary.com* – zawód uznany w 2006 roku za najlepszy w Stanach Zjednoczonych, pod względem możliwości rozwoju, wysokości pensji, poziomu stresu, elastyczności godzin pracy i środowiska pracy
- wg Careercast: 11 miejsce na liście najmniej stresujących zawodów w 2019 r. (spokojna praca przed komputerem z bardzo dobrymi perspektywami zatrudnienia)
- <https://www.careercast.com/jobs-rated/2019-jobs-rated-report>



Software Developer

Overall Rating: 11/200

Median Salary: \$103,560

Work Environment

Stress

Projected Growth

Good

Very Low

Very Good

68/200

27/200

22/200

[View Raw Scores](#)

[Search Software Developer Jobs](#) ➔



POLSKO-JAPOŃSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Stanowiska pracy

<https://news.codecademy.com/what-does-a-software-engineer-do/>

- Software engineer
- Project manager
- Database architect
- Software architect
- Project manager
- Team leader
- Data Scientist
- Programista ORACLE, SQL, .NET etc.
- Administrator hurtowni danych
- Wszelkie inne rodzaje (z danych poniżej ok. 36000 ofert pracy na całym świecie)



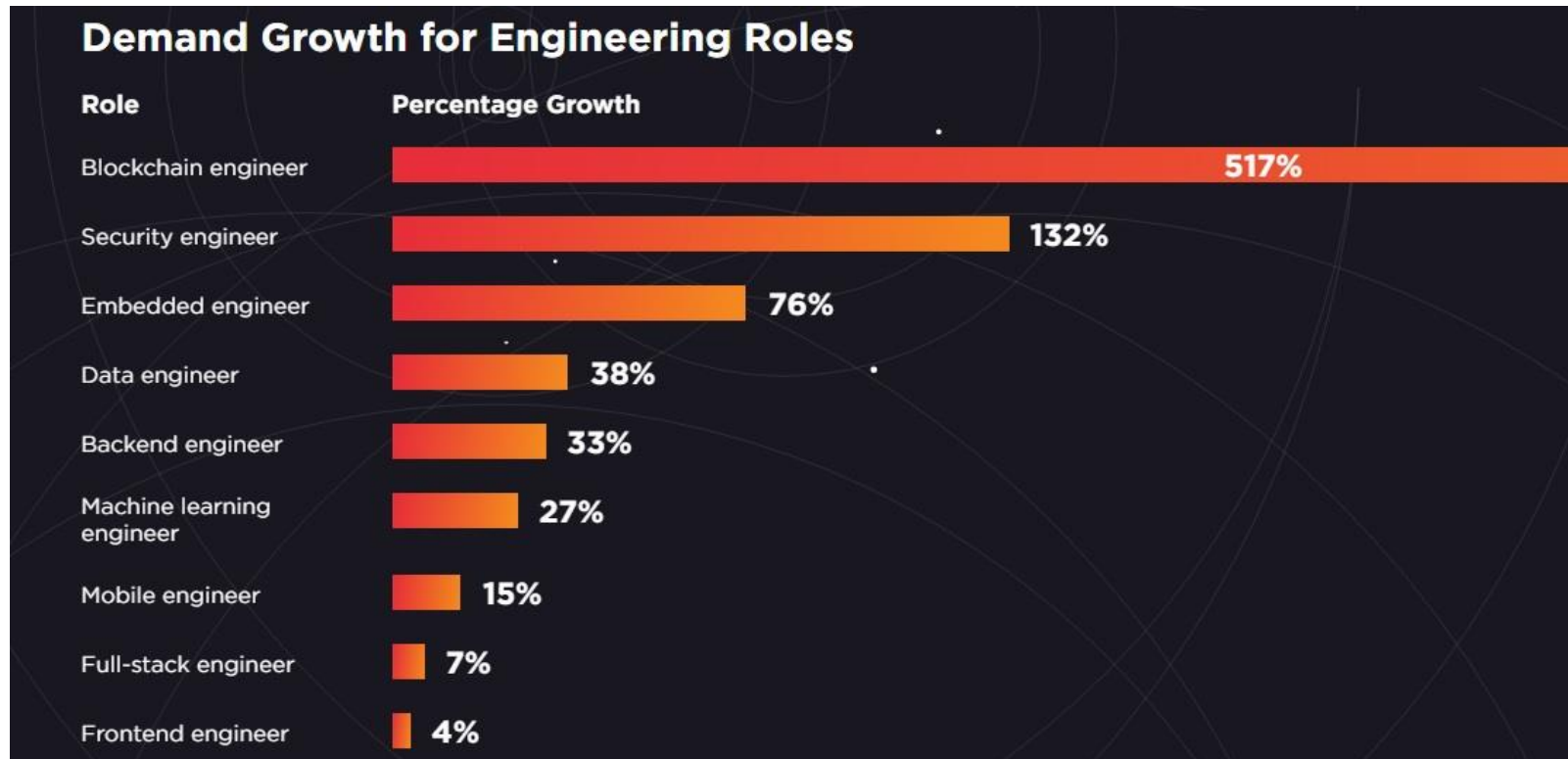
Polecam dla zainteresowanych:

https://www.careercast.com/jobs/search/results?isFbWidget=false&kwsPrimary=software%20engineer&location=United%20States&location=&autocomplete_location=&type=jobsearch&type=jobsearch&widgetref=&widgetref=&widget=1&widget=1



POLSKO-JAPOŃSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Software engineer jobs



<https://www.houseofbots.com/news-detail/11583-4-9-hottest-software-engineering-jobs-in-demand-&-the-high-salaries-they-command-across-world>

A Day in the Life of a Software Engineer

Entry-Level Software Engineer



Implement
new
features

React
Ruby
Python

learn new
languages &
frameworks



Maintain
existing code



Pair
programming



Resolve simple
bugs / errors



Refactor code

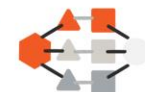


Practice test driven
development (TDD)

Decide which
framework &
libraries to use



Architect API
endpoints &
data models



Mentor junior
engineers



Attend sprint planning
meetings with product
managers & designers



Code
reviews



Refactor legacy code



Architect scalable
systems



Senior Software Engineer



Faza przedprojektowa

1. Planowanie.
2. Faza Strategiczna.



**Skąd impuls inicjujący nowe projekty
lub będące kontynuacją?**



POLSKO-JAPONSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

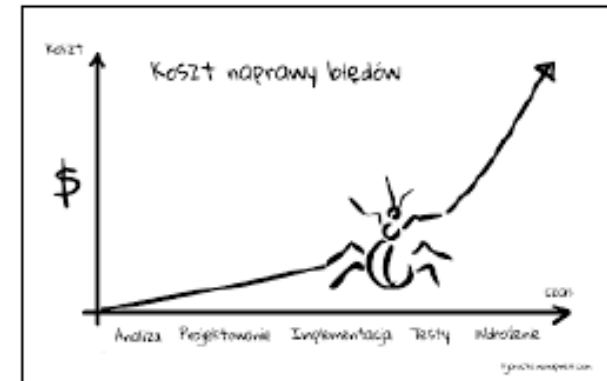
Główne czynności w fazie przedprojektowej

- ☐ identyfikacja i opis problemu
- ☐ sformułowanie wizji rozwiązania
- ☐ ocena zagrożeń i ewentualnie
analiza (studium) wykonalności
- ☐ decyzja, inicjowanie i ustanowienie projektu
- ☐ opracowanie wstępnego plan projektu

Głównie po stronie Zamawiającego!

MOTYWACJA

- Skutki błędów, koszty naprawy
 - przyjmuje się, że koszt zmian i naprawy błędów wzrasta 10x z każdą kolejną fazą projektu
- Stąd wszystkie praktycznie metodyki zakładają działania przedprojektowe



podejścia tradycyjne : *planowanie, faza przedprojektowa*

podejście agile : *gra planistyczna*

metodyka PRINCE2 : *Przygotowanie projektu
oraz Inicjowanie projektu*

metodyka PCM (Project Cycle Management)
: *analiza interesariuszy, analiza problemów,
analiza celów oraz analiza strategii*

Identyfikacja problemu

➤ Sytuacja problemowa

- Usytuowanie biznesowe
- Kontekst
- Zakładane cele do osiągnięcia i spodziewane korzyści
- Udziałowcy

➤ Aktualny system

- Istniejący system (także jego działanie)

➤ Na czym polegają problemy

➤ Istniejące rozwiązania (wzorce)

- Także konkurencyjne, ich plusy i minusy





POLSKO-JAPONSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Metody opisu problemu

➤ Opis nieformalny

- często zawiera schematy, rysunki

➤ Opis ustrukturalizowany

➤ Soft Systems Methodology (SSM, społeczna metoda budowy systemów)

Checkland P., and Scholes J.: Soft Systems Methodology in Action.
J. Wiley & Sons, 1990

Szejko St. (red): Metody wytwarzania oprogramowania, MIKOM 2002

Wrycza St.: Analiza i projektowanie systemów informatycznych zarządzania.
Metody, techniki, narzędzia. PWN, Warszawa 1999

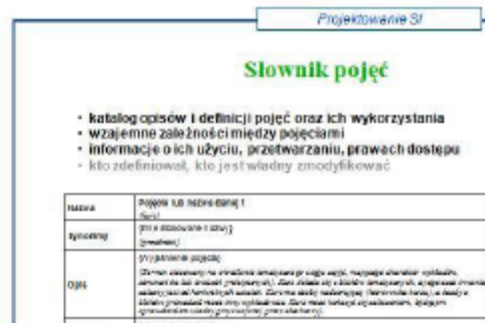
Jakie sposoby są najlepsze?

Ustrukturalizowany opis problemu

- Definicja problemu

Element	Opis
Problem polega na ...	Opisz problem
Problem dotyczy ...	Zidentyfikuj udziałowców, których dotyczy problem
Rezultatem problemu jest ...	Opisz wpływ tego problemu na udziałowców i działalność przedsiębiorstwa.
Korzyści z rozwiązania problemu ...	Wskaż proponowane rozwiązanie i wymień podstawowe korzyści

- Słownik pojęć
- Wsparcie diagramami, schematami,...





POLSKO-JAPONSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Wzbogacony wizerunek (Rich Picture)

Soft Systems Methodology Checklanda

Na początku przedsięwzięcia projektowego zainteresowane osoby mają bardzo nieostre wyobrażenie o aktualnej sytuacji, potrzebach i docelowym kształcie tego, co ma zostać osiągnięte. Nawet jeśli elementy te można określić, konieczne jest wyrażenie ich w sposób czytelny zarówno dla analityka systemu jak i potencjalnych wykonawców.

Użyteczną techniką jest utworzenie *Rich Picture* - *Wzbogaconego wizerunku*. Wizerunek taki odzwierciedla zagadnienia składające się na kontekst i działanie organizacji, sytuację problemową, uwarunkowania.

Wzbogacony Wizerunek stanowi istotną pomoc dla analityka systemu, jako że zapewnia całościowe spojrzenie na obszar problemu, wymuszając większe jego zrozumienie. **Konieczność 'zgrania' punktów widzenia wielu udziałowców i wyrażenia ich na niewielkim obszarze wymusza ostrość spojrzenia na problem.**



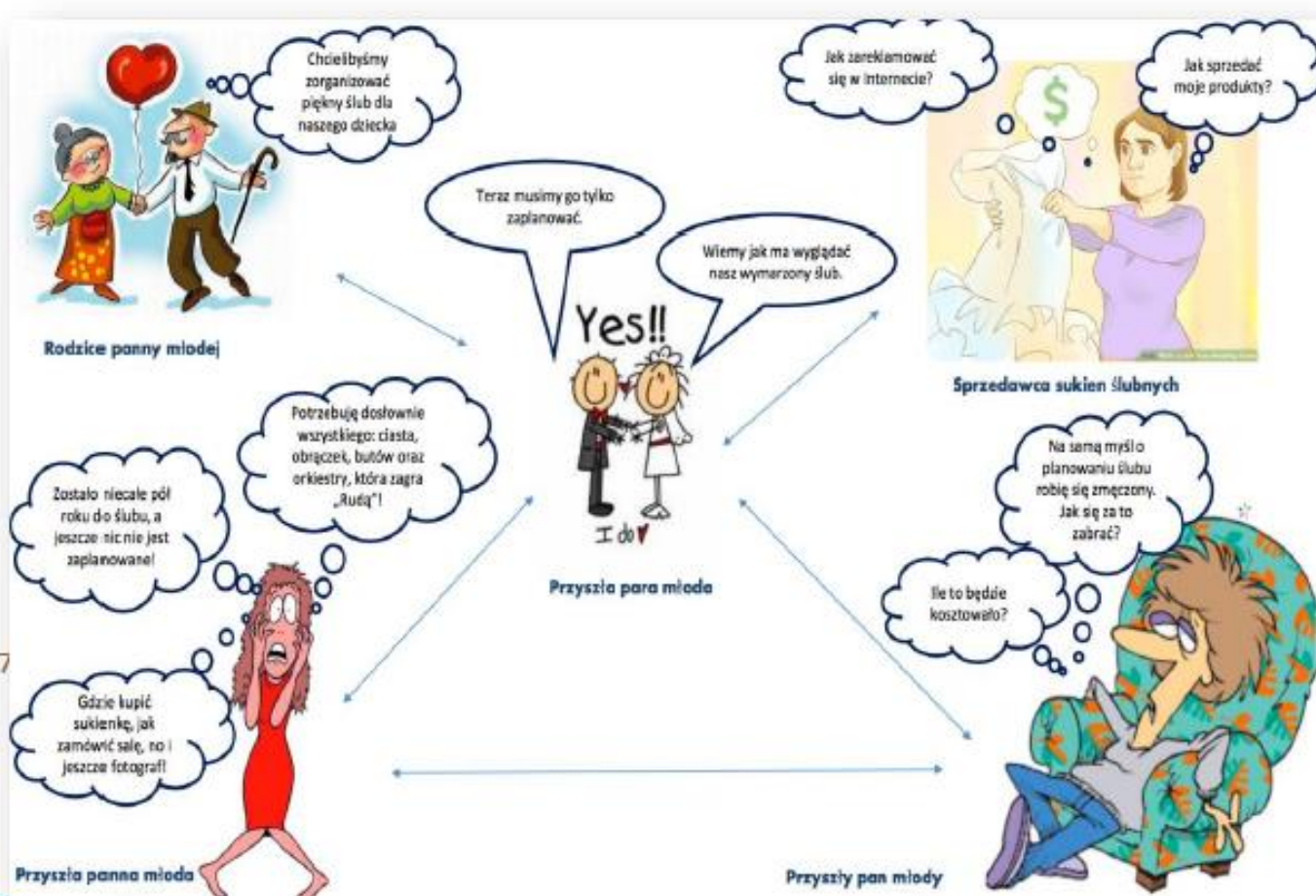
Aspekty problemu ujmowane na Rich Picture

Soft Systems Methodology Checklanda

- udziałowcy i elementy struktury obszaru problemowego (może to być dekompozycja na wydziały, fizyczne lub geograficzne umiejscowienie, jednostki działające i współdziałające),
- zachodzące procesy, czyli działania, jakie mają miejsce w systemie oraz związki pomiędzy elementami wizerunku
- związki pomiędzy elementami struktury i zachodzącymi procesami stanowiące esencję obszaru problemowego - będą one odzwierciedlać konflikty, obawy, zagrożenia, nieporozumienia pomiędzy nowymi procesami a starymi strukturami

Wzboagacony wizerunek (Rich Picture)

Wzboagacony obrazujący
problemy udziałowców
napotykanne
przy planowaniu ślubu
M. Niegrzybowska, G. Mrósk,
M. Piekarska, A Borkowski
Aplikacja wspomagająca
organizację ślubów i wesel
„AJ-DU”. Praca dypl. PJATK, 2017

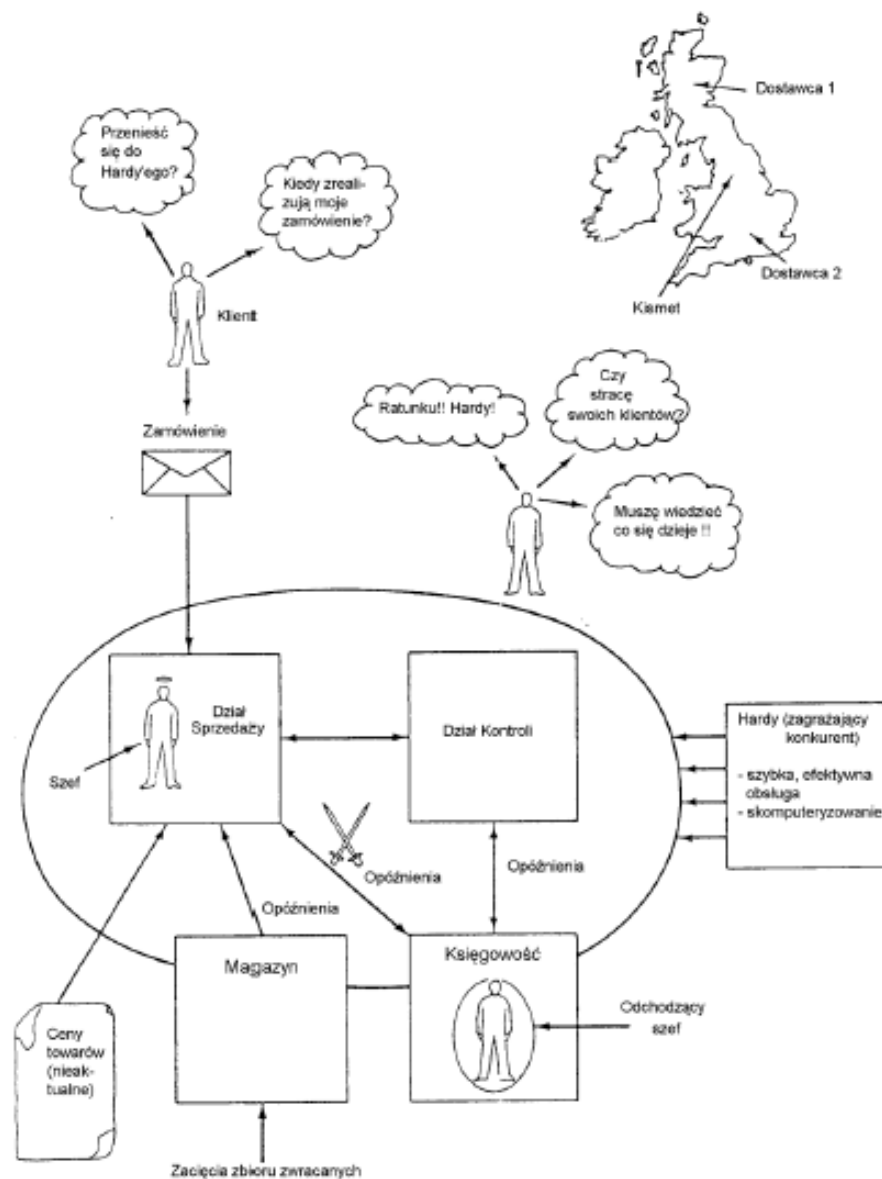


Rich Picture

Przykład Kismet Ltd. - hurtowego dostawcy sprzętu el

Przedsiębiorstwo Kismet zostało założone 30 lat temu przez seniora rodu i przez długi czas rozwijało się bardzo dynamicznie. W ciągu ostatnich 3 lat Mr. Kismet (junior) uzyskał 300% wzrost obrotów handlowych. Pojawiły się też problemy. Dominacja Kismetu na rynku w północno-wschodniej części kraju jak i ekspansja na rynek północno-zachodni została poważnie zagrożona przez konkurenta: Hardy Ltd. Ten ostatni został założony w ciągu ostatnich 9 m-cy z dużym zastrzykiem kapitału. Za pomocą bezpośredniej linii telefonicznej i Internetu umożliwia on klientom korzystanie z odpowiednich kart kredytowych przy zamawianiu sprzętu, i coraz bardziej zagraża Kismet Ltd na rynku hurtowej sprzedaży sprzętu elektronicznego.

Od pewnego czasu zarządzający Kismet Ltd zdają sobie sprawę z bardzo wolnego przetwarzania otrzymywanych zamówień. Zużywany czas wydłuża się coraz bardziej, co jeszcze spotęgował rozwój firmy w ciągu ostatnich trzech lat. Całkowicie ręczna obróbka danych 'nie radzi sobie' z dużą ilością informacji, mimo zwiększenia zatrudnienia. Natomiast Hardy jest w stanie zaoferować klientom szybką i efektywną obsługę - posiada system informatyczny ze skomputeryzowanym przetwarzaniem danych



Rich Picture

Rysowanie powinno:

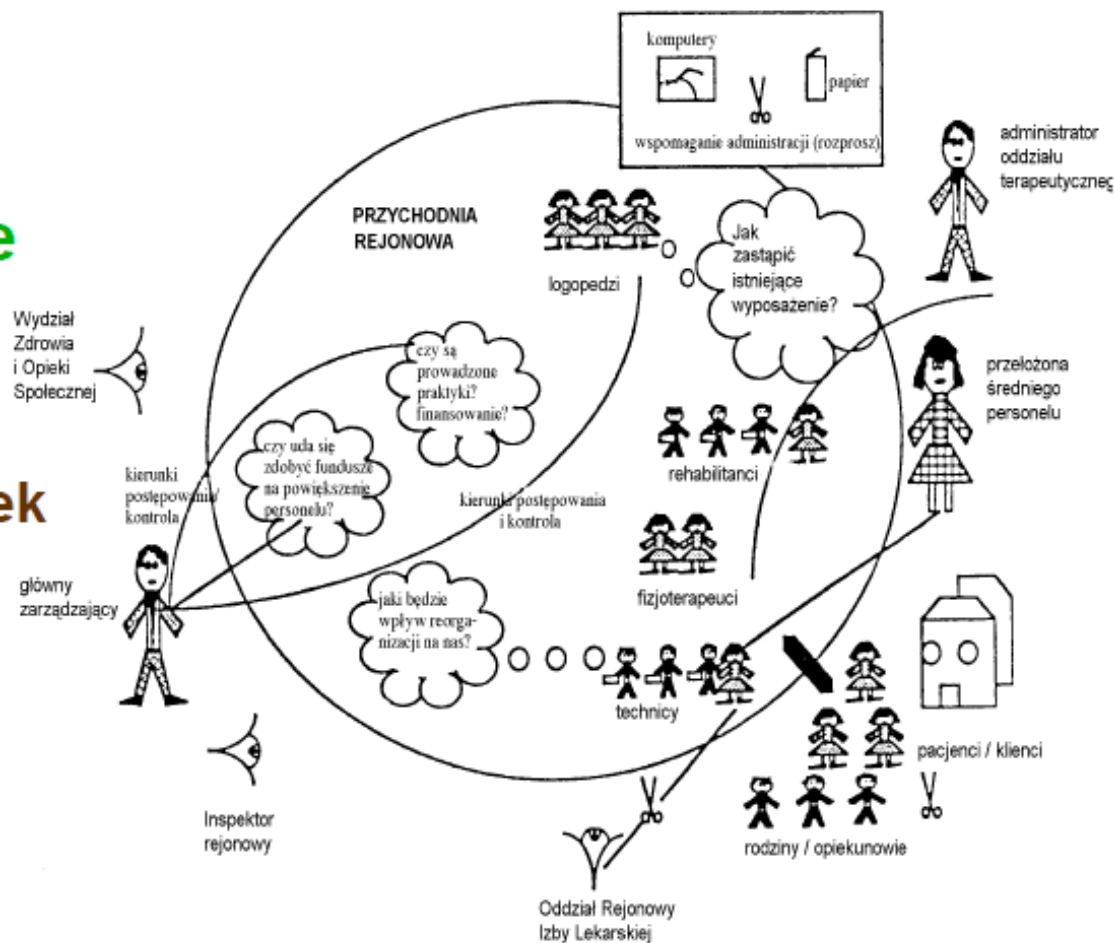
- **skupiać uwagę na najważniejszych zagadnieniach.** Częstym błędem analizy jest zbyt głębokie wnikanie w szczegóły, zaciemniające obraz na tym etapie prac. To, co jest nieodzowne w dalszych krokach analizy i konstrukcji systemu oraz wspierane przez odpowiednie po temu techniki i narzędzia, na wstępnym etapie prac może spowodować, iż analityk nie będzie widział *'lasu wśród drzew'*;
- pomóc wszystkim uczestnikom **określić rolę, jaką pełnią oni w działaniu organizacji.** Analityk może wytworzyć sobie niewłaściwy, fragmentaryczny lub subiektywny model jej działania - niezgodności takiego modelu są łatwiejsze do wychwycenia gdy nada się mu formę graficzną;
- wzbogacony wizerunek może być wykorzystany do **określenia tej części organizacji, która będzie podlegać informatyzacji;**
- wzbogacony wizerunek może być środkiem, służącym do **wyrażenia obaw i odpowiedzialności pracowników, jak również konfliktów** zachodzących pomiędzy zaangażowanymi osobami.

Nie ma formalnej techniki rysowania Rich Picture. Ale ...

Wzbogacony wizerunek

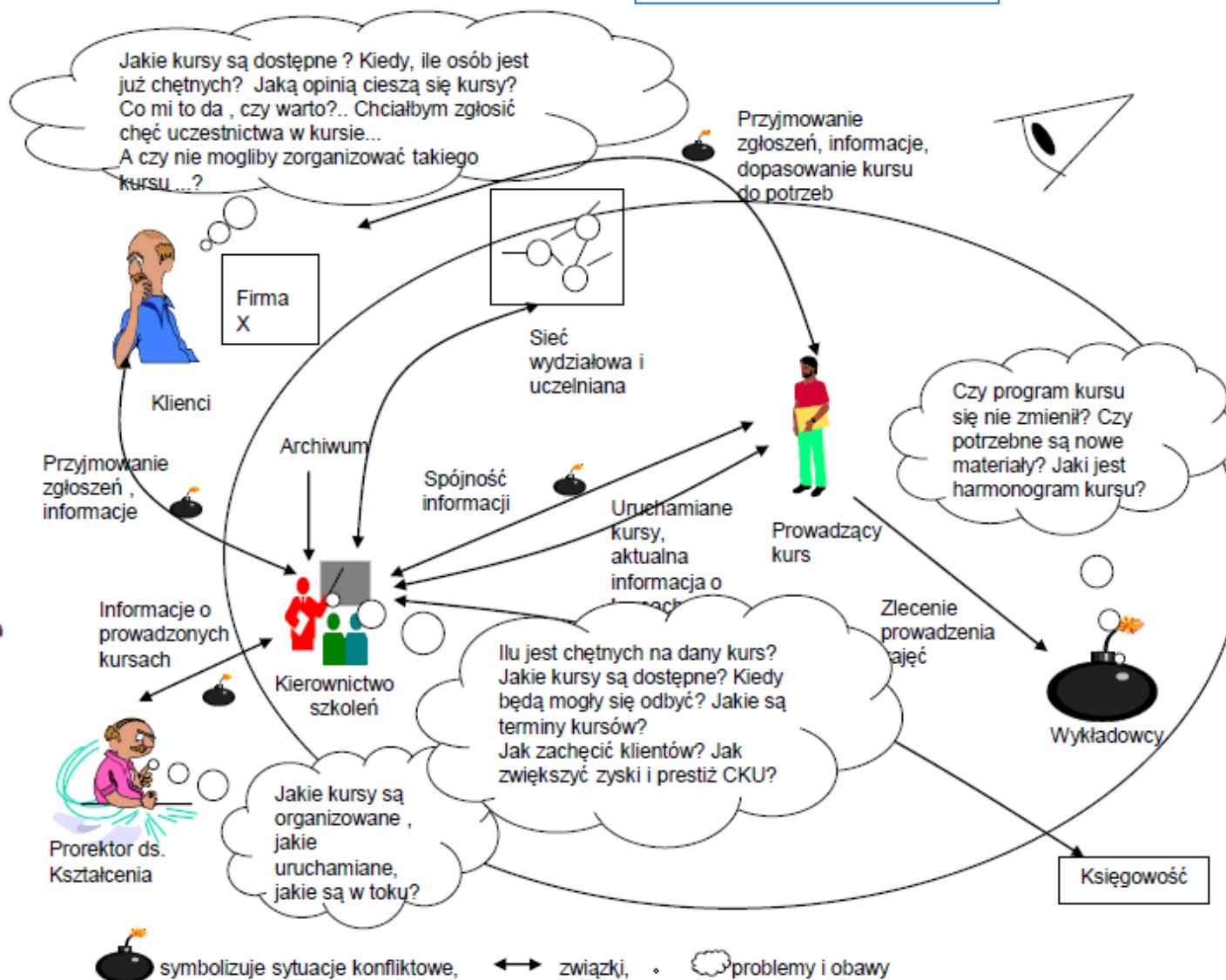
Wzbogacony
wizerunek –
standardowo
wykorzystywane
symbole

Przykład: Wizerunek
przychodni
terapeutycznej



Wzbogacony wizerunek

Wzbogacony wizerunek Przykład CKU



Wzbogacony wizerunek (Rich Picture)

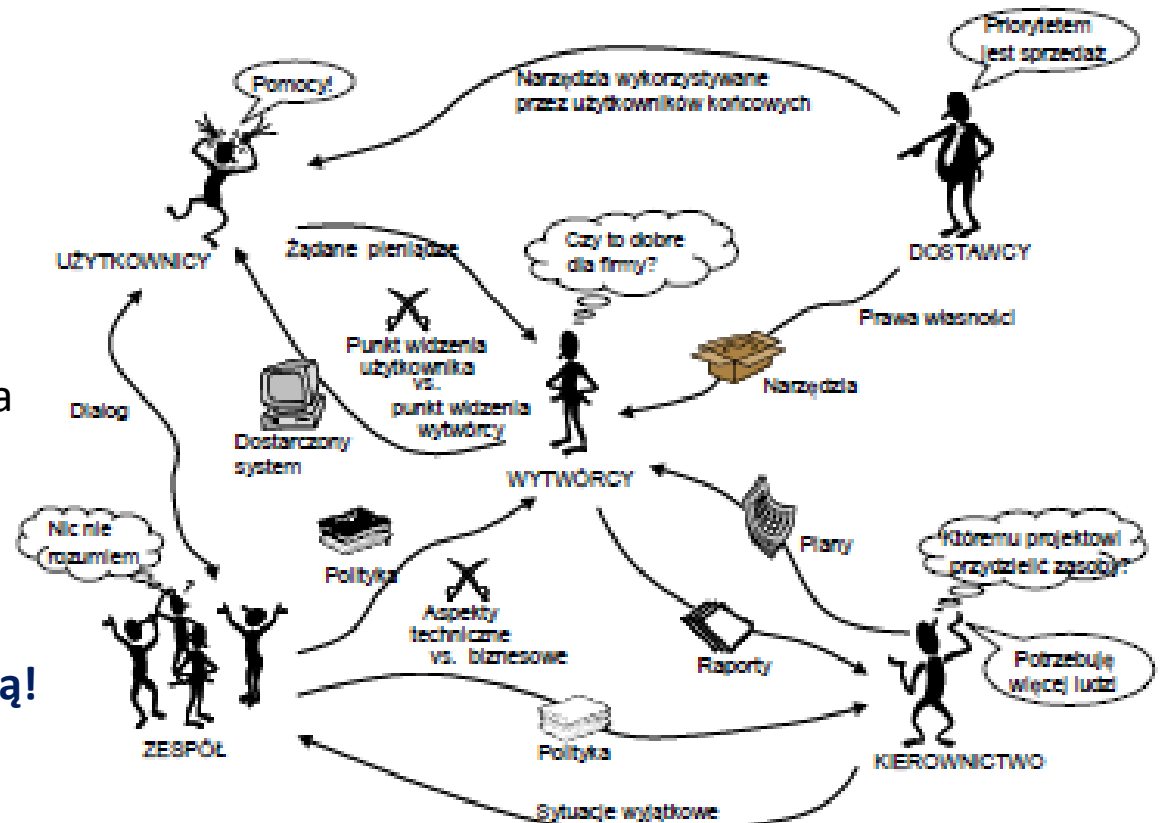
UWAGA!

Wzbogacony wizerunek

NIE

służy do opisu wizji rozwiązania

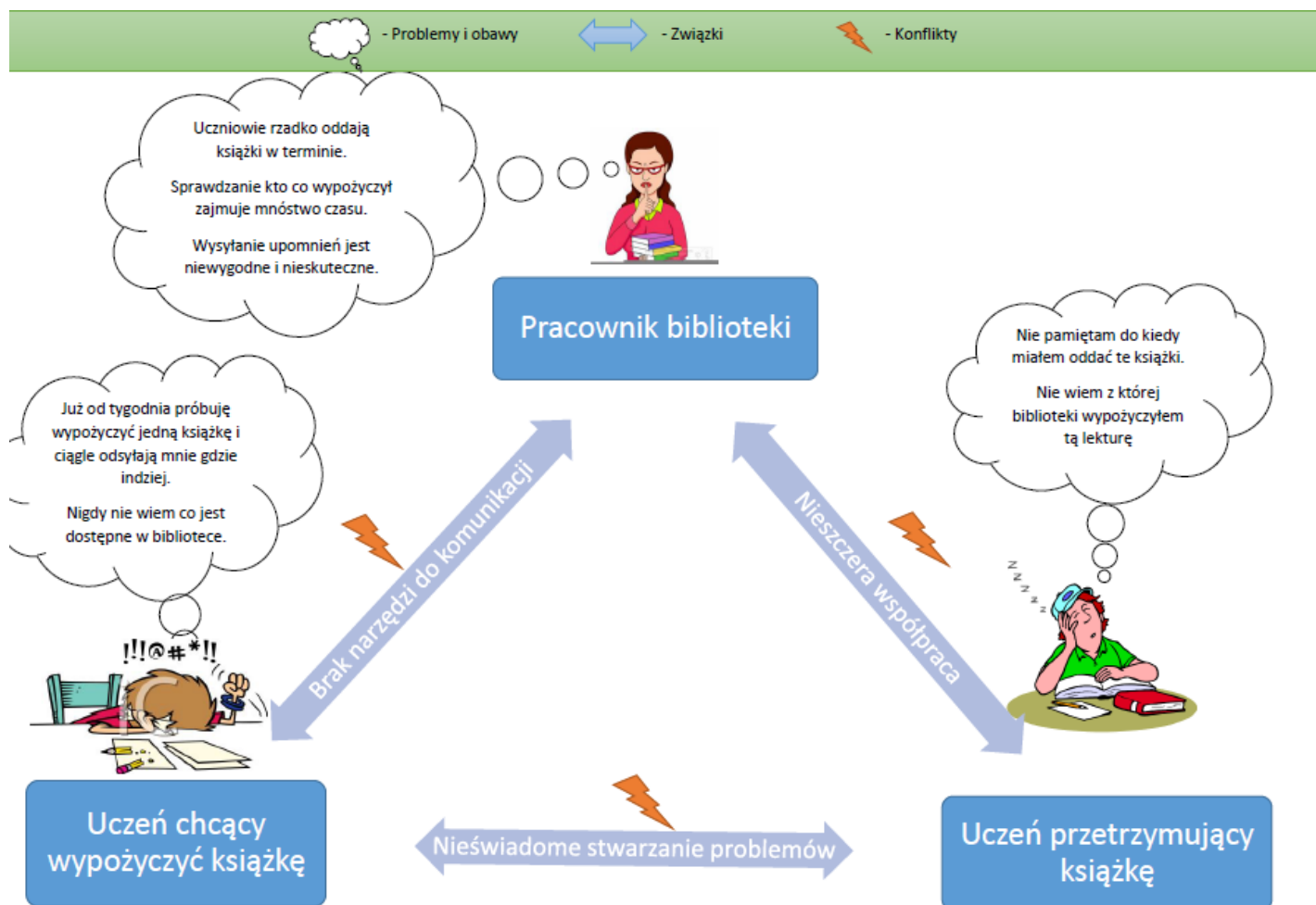
Definiuje sytuację problemową!



Wzbogacony wizerunek Wytwarzania oprogramowania

Wzbogacony wizerunek (Przykłady)

E-wypożyczalnia



Wzbogacony wizerunek (Przykłady)

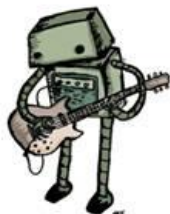
Nauka i zabawa -music

Muzyka jest fajna, wszyscy ją słuchają, wielu chciałoby nauczyć się grać na jakimś instrumencie - np gitarze

Do wyboru jest wiele instrumentów, ale najpopularniejsze to gitary (akustyczna, basowa, elektryczna), perkusja, fortepian, skrzypce.



≠



W efekcie muzyk uczy się mechanicznego odtwarzania ruchów zamiast prawdziwej umiejętności gry na instrumencie

=

W efekcie, początkujący muzycy zamiast uczyć się notacji muzycznej, wolą korzystać z tabulatur

wersja 1

e		---	---	---	---	---	---
B		---	---	---	---	---	---
G		---	---	---	---	---	---
D		---	---	---	---	---	---
A		---	---	---	---	---	---
E		---	---	---	---	---	---

wersja 2 na jednej strunie basowej

e		---	---	---	---	---	---
B		---	---	---	---	---	---
G		---	---	---	---	---	---
D		---	---	---	---	---	---
A		---	---	---	---	---	---
E		---	---	---	---	---	---



- * narzuca sposób gry autora tabulatury
- * pisana jest na jeden konkretny instrument
- * nie uwzględnia dokładnego zapisu rytmicznego
- * narzuca miejsce zagrania dźwięku

A do zapisu muzyki na wszystkie instrumenty stosuje się notację muzyczną! Nie ważne jaki instrument wybierzesz, utwory będą prawie zawsze zapisywane w ten sam sposób!



lenistwo i szukanie drogina skróót



W tym miejscu jednak powstają pewne trudności które zniechęcają początkujących muzyków



Skomplikowany zapis jest trudny i zniechęcający w nauce.



Wymaga nakładów pieniężnych na lekcje lub materiały do nauki

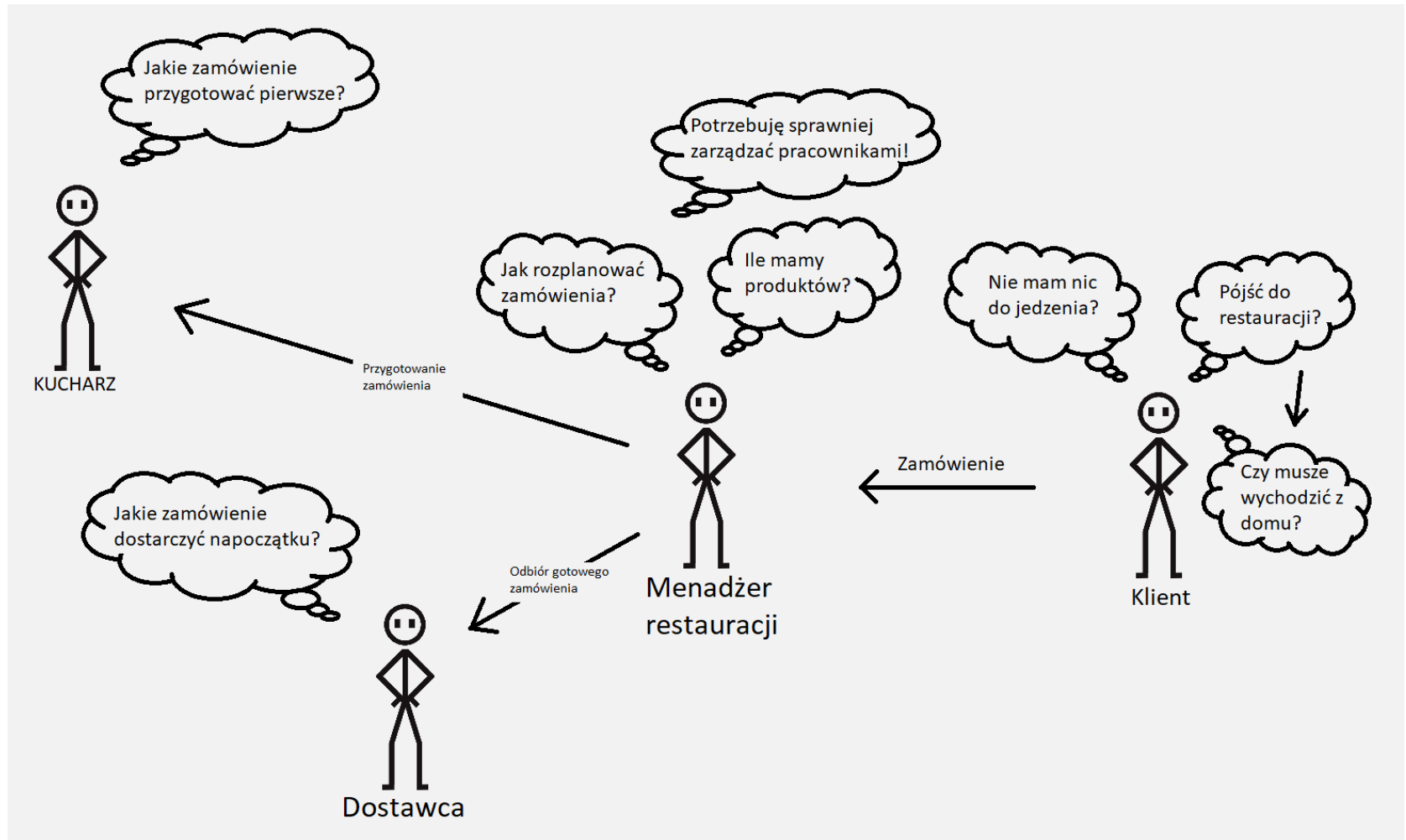


Nauka jest żmudna i czasochłonna



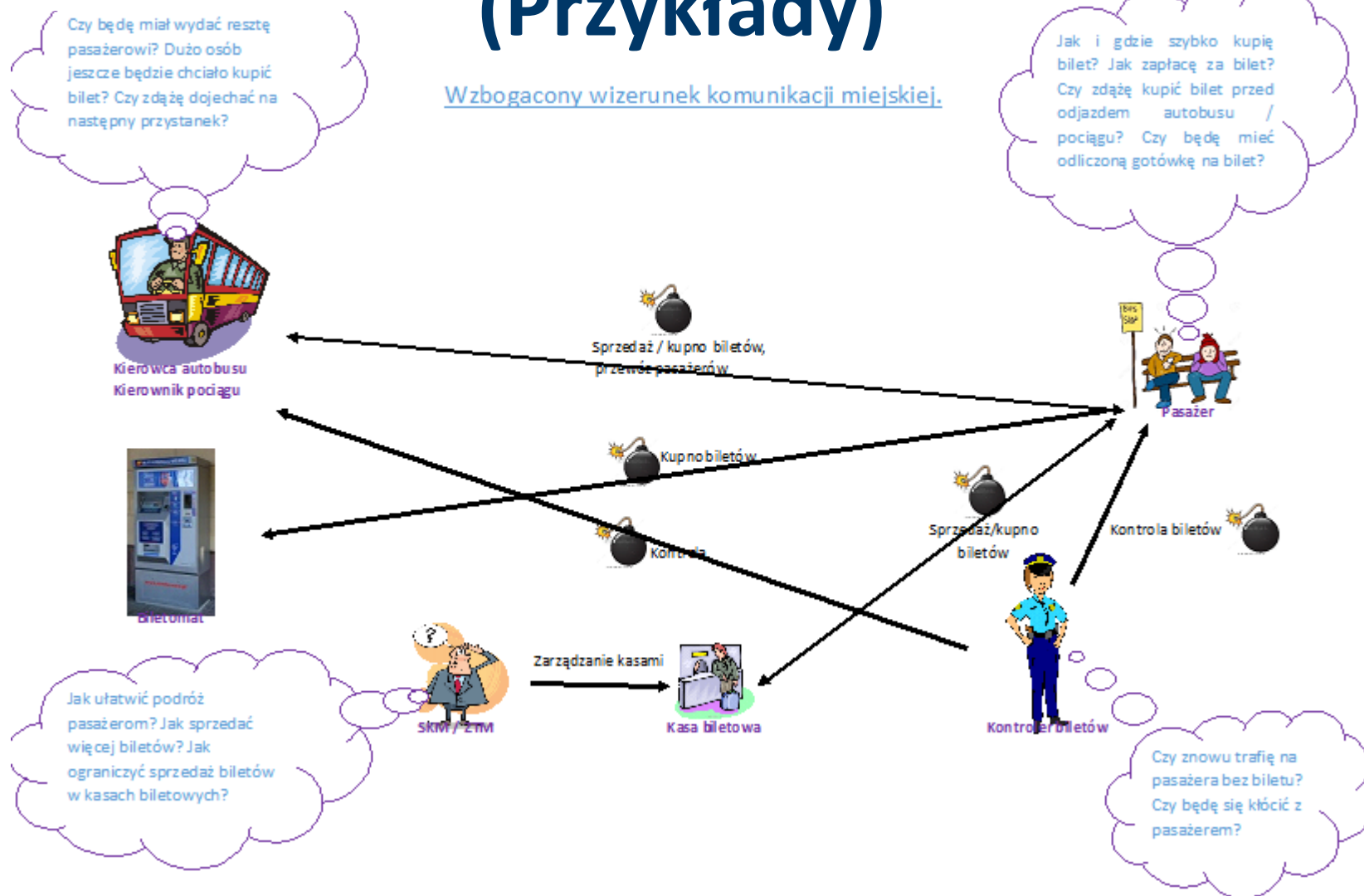
Wzbogacony wizerunek (Przykłady)

Restaurant management system



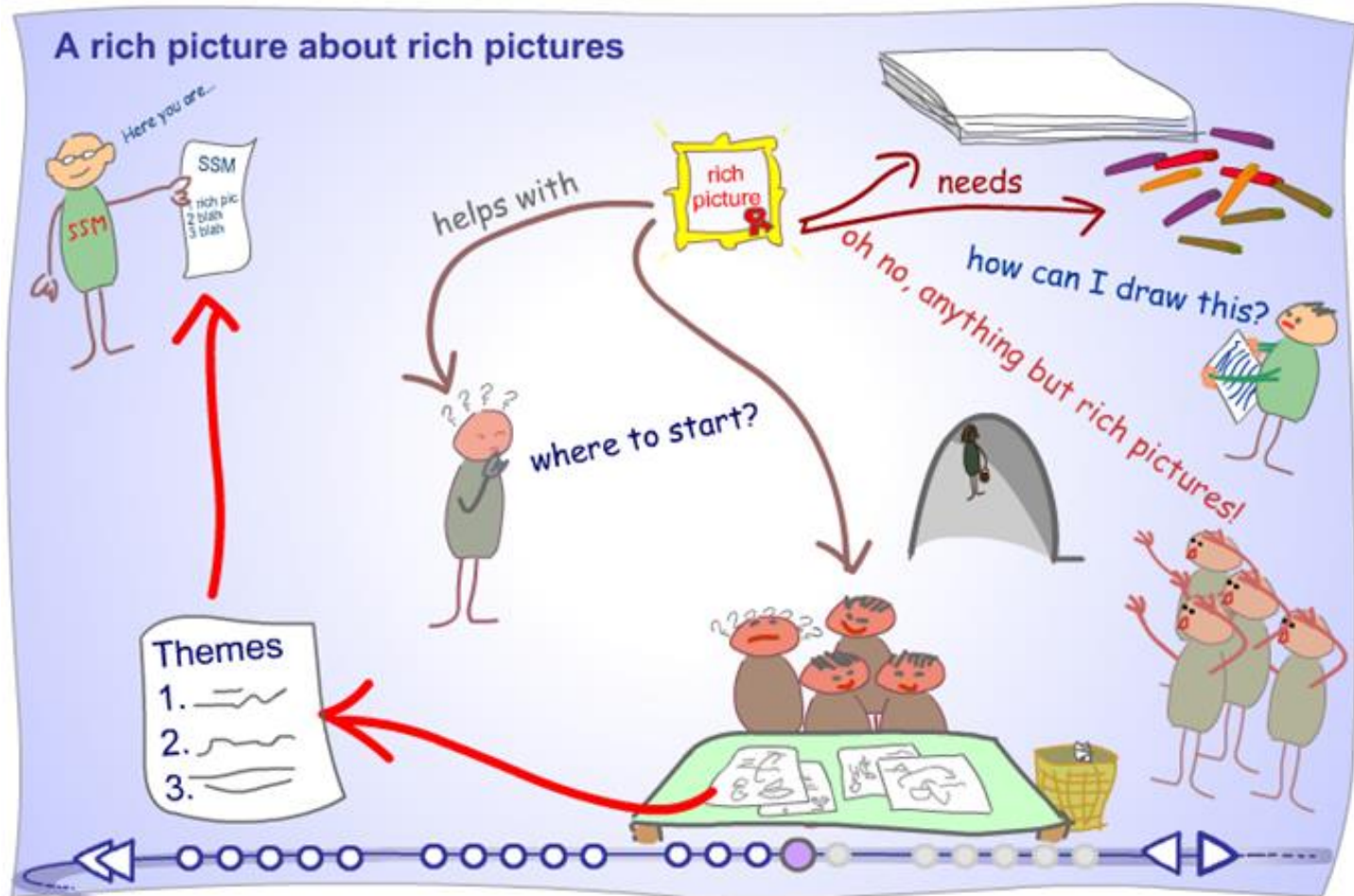
Wzbogacony wizerunek (Przykłady)

Wzbogacony wizerunek komunikacji miejskiej.



Wzbogacony wizerunek (Przykłady)

Rich picture o rich picture





POLSKO-JAPOŃSKA
AKADEMIA TECHNIK
KOMPUTEROWYCH

Pytania i działania

Co system ma robić?

W jaki sposób ma działać?

Realizacja:

Project pitch

Wzbogacony wizerunek

Karta projektu

Dokument Założeń Wstępnych