Wstęp do analizy biznesowej

Wykład 02

Analiza i projektowanie systemów

dr Monika Sitarska-Buba

Źródło:

Business Analysis, Fourth Edition, By: James Cadle, Debra Paul - 8h 8m8hours 8minutes, Publisher: BCS © 2020

***Requirements Analysis and System Design*, 3rd ed.,** MACIASZEK, L.A. (2007), Addison Wesley, Pearson, Harlow England, 642p., ISBN 978-0-321-44036-5
http://www.comp.mq.edu.au/books/rasd3ed/

Tematyka

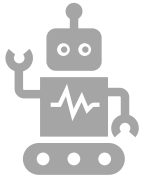
1. Czym jest analiza biznesowa?
2. Definiowanie wymagań i ich kategorie
3. Gromadzenie wymagań
4. Dokumentowanie wymagań
5. Negocjacja i walidacja wymagań

1. Czym jest analiza biznesowa?

- Definicje i pojęcia powiązane
- Role w projekcie IT

Pojęcie analizy biznesowej

Według formalnej definicji **IIBA** (International Institute of Business Analysis) analiza biznesowa to „określanie potrzeb i rekomendowanie rozwiązań”.



BABOK (Business Analysis Body of Knowledge) **2.0**:

jest to zestaw czynności wykonywanych w budowaniu relacji między interesariuszami po to, aby zrozumieć strukturę, polityki (interesy), procesy danej organizacji i zaproponować takie rozwiązania, dzięki którym organizacja będzie realizować swoje cele.

BABOK 3.0:

są to działania praktyczne umożliwiające zmianę w przedsiębiorstwie poprzez zdefiniowanie potrzeb i rekomendowanie rozwiązań, które dostarczą wartości interesariuszom.

Analiza biznesowa a analityka biznesowa

Specjalista Business Intelligence skupia się na gromadzeniu, transformacji i wizualizacji danych w celu wspierania procesów podejmowania decyzji i podnoszenia wydajności przedsiębiorstwa.

Analityk biznesowy natomiast definiuje potrzeby i tworzy rozwiązania, które na nie odpowiadają. Praca analityka biznesowego jest nierozerwalnie związana z procesem zmiany, a jej zakres może wykraczać poza systemy informatyczne.

Role w projektach IT

- **Sponsor** – osoba, która chce zmiany w organizacji, finansuje tę zmianę I odpowiada za jej rezultat (płaci za stworzenie system, żeby przyspieszyć procesy, zwiększyć wydajność w celu optymalizacji zysku)
- **Interesariusz** – osoby, które skorzystają dzięki zmianie w organizacji (np. dzięki automatyzacji ich praca zostanie przyspieszona)
- **Project Manager** – osoba, która odpowiada za organizację I sprawny przepływ pracy w projekcie
- **Zespół projektowy -** developerzy, testerzy, architekci, UX/UI designerzy

Analityk Biznesowy

- **Analityk biznesowy** to **osoba, który łączy biznes i IT**, aby tworzyć rozwiązania wprowadzające zmianę i dające wartość dla organizacji.
- Aby dostarczyć wartość i sprawnie przeprowadzić zmianę, analityk może współpracować między innymi z dyrektorami, managerami, jak i specjalistami różnych szczebli.
- Aby stworzyć faktycznie działające rozwiązanie, analityk musi współpracować z developerami, testerami czy project managerami.
- Jego rolą jest dostarczenie wskazówek, pomagających stworzyć system, realizujący potrzebę biznesową klienta.

2. Definiowanie wymagań i ich kategorie

- Wymagania biznesowe
- Wymagania użytkownika (interesariuszy)
- Wymagania systemowe

Wymagania biznesowe

Opisują korzyści biznesowe, jakie organizacja chce osiągnąć. Wymagania biznesowe skupiają się na celach biznesowych.

W treści wymagania biznesowego nie powinno być elementów systemowych, lecz cel biznesowy.

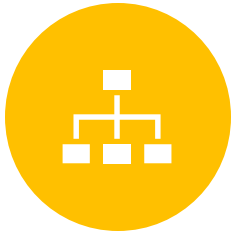
W celu zidentyfikowania wymagania biznesowego zastanów się wspólnie z Biznesem: “Co organizacja chce osiągnąć?”.

Przykład: “Skrócenie czasu realizacji zamówienia o 10% od stanu obecnego”

Wymagania użytkowników (interesariuszy)

Opisują cele oraz korzyści biznesowe, jakie osiągną użytkownicy systemu.

Opisują co użytkownicy będą mogli robić za pomocą systemu.



Często można spotkać się z określeniem: wymagania biznesowe interesariuszy.

Przykład: “Otrzymanie powiadomienia o nowym zamówieniu do realizacji przez Magazyn”.

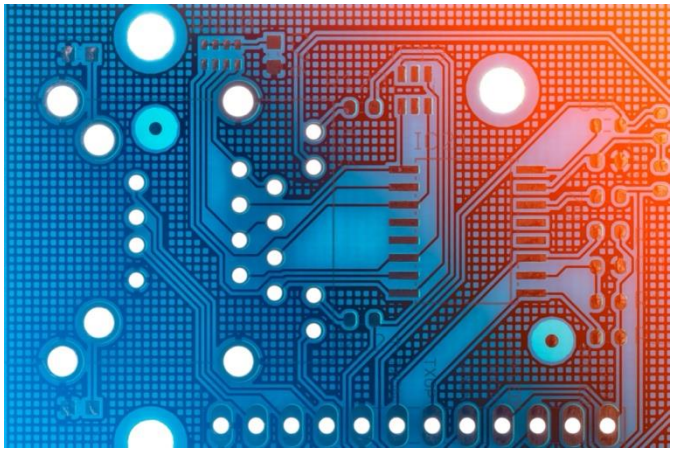
Wymagania systemowe

wymagania funkcjonalne – opisują zachowania opisywanej funkcji rozwiązania, które realizują wymagania biznesowe. W celu łatwiejszej identyfikacji wymagania funkcjonalnego zadaj sobie pytanie: “Co funkcja musi umieć robić?”

Pozafunkcjonalne (niefunkcjonalne) – są dopełnieniem wymagań funkcjonalnych opisując cechy rozwiązania pod kątem jakościowym, np. wydajności czy niezawodności. Pomocnym pytaniem podczas identyfikacji tych wymagań: “Jako dobrze system ma to robić? W jaki sposób?”.

Kategorie wymagań niefunkcjonalnych wg ISO 25010

Źródło: http://www.cs.put.poznan.pl/jrojek/files/io1/requirements/NFR_kategorieISO25010.pdf



Jakość produktu

FUNKCJONALNE DOPASOWANIE (ANG. FUNCTIONAL SUITABILITY) - STOPIEŃ, W JAKIM ZBIÓR FUNKCJI POKRYWA WSZYSTKIE OKREŚLONE ZADANIA I CELE.

FUNKCJONALNA POPRAWNOŚĆ - STOPIEŃ, W JAKIM PRODUKT DOSTARCZA POPRAWNE REZULTATY Z WYMAGANĄ PRECYZJĄ.

FUNKCJONALNA ODPOWIEDNIOŚĆ - STOPIEŃ, W JAKIM FUNKCJE PROGRAMU ULATWIAJĄ OSIĄGNIĘCIE OKREŚLONYCH ZADAŃ I CELÓW.

Wydajność (ang. Performance efficiency)

CHARAKTERYSTYKA CZASOWA - STOPIEŃ, W JAKIM CZAS ODPOWIEDZI I CZAS PRZETWARZANIA ORAZ PRZEPUSTOWOŚĆ SPEŁNIAJĄ WYMAGANIA.

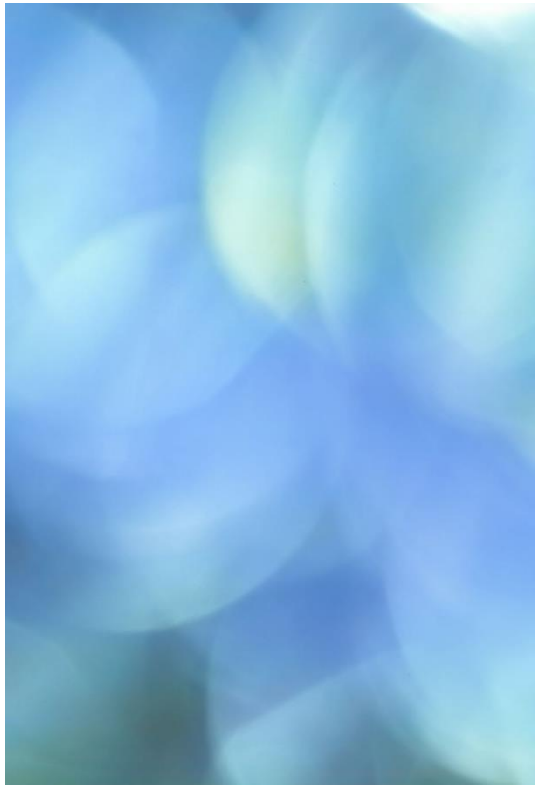
ZUŻYCIE ZASOBÓW - STOPIEŃ SPEŁNIENIA WYMAGAŃ DOTYCZĄCYCH ILOŚCI I TYPÓW ZASOBÓW WYKORZYSTYWANYCH PRZEZ PRODUKT LUB SYSTEM PODCZAS WYKONYWANIA JEGO FUNKCJI.

OCZEKIWANA WYDAJNOŚĆ (ANG. CAPACITY) - STOPIEŃ, W JAKIM SĄ SPEŁNIONE MAKSYMALNE OGRANICZENIA NAŁOŻONE NA PARAMETRY PRODUKTU LUB SYSTEMY W WYMAGANIACH.

Kompatybilność (ang. Compatibility)

Współistnienie - stopień, w jakim produkt może skutecznie realizować wymagane od niego funkcje podczas współdzielenia z innymi produktami wspólnego środowiska i zasobów bez szkodliwego wpływu na jakikolwiek inny produkt.

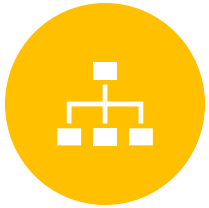
Interoperacyjność - stopień, w jakim dwa lub więcej systemów, produktów lub komponentów mogą wymieniać dane i przetwarzać je.



Użyteczność (ang. Usability)

Rozpoznawalność zastosowania - stopień, w jakim użytkownicy mogą rozpoznać czy produkt lub system odpowiada ich potrzebom.

Łatwość nauczenia się - stopień, w jakim produkt lub system może być wykorzystywany przez określonych użytkowników w celu osiągnięcia określonych celów związanych z nauczeniem się korzystania z produktu lub systemu.



Łatwość operowania - stopień, w jakim produkt lub system umożliwia łatwe operowanie I sterowanie nim.

Ochrona użytkownika przed błędami - stopień, w jakim system chroi użytkowników przed popełnieniem błędów.

Estetyka interfejsu użytkownika - stopień, w jakim interfejs użytkownika pozwala użytkownikowi na miłą i satysfakcjonującą interakcję.

Dostępność personalna - stopień, w jakim produkt lub system może być wykorzystywany przez ludzi o bardzo różnych cechach i możliwościach do osiągnięcia określonego celu wokreślonym kontekście użycia.

Niezawodność (ang. Reliability)

Dojrzałość - stopień, w jakim system, produkt lub komponent spełnia wymagania niezawodności w normalnyc warunkach pracy.

Dostępność techniczna - stopień, w jakim system jest sprawny i dostępny kiedykolwiek jest potrzebny.

Odporność na wady - stopień, w jakim system, produkt lub komponent działa zgodnie z przeznaczeniem pomimo wad sprzętu lub oprogramowania.

Odtwarzalność - stopień, w jakim w przypadku awarii lub przerwania produkt lub system może odtworzyć dane bezpośrednio naruszone przez tą awarię i przywrócić pożądany stan systemu.

Bezpieczeństwo (ang. Security)

Poufność - stopień, w jakim produkt lub system zapewnia, że dane są dostępne tylko dla upoważnionych osób.

Integralność - stopień, w jakim system, produkt lub component zapobiega nieautoryzowanej modyfikacji programu komputerowego lub danych.

Niezaprzeczalność - stopień, w jakim można udowodnić podjęcie działań lub wystąpienie zdarzeń tak, by nie można było zaprzeczyć ich wystąpieniu.

Identyfikowaloość - stopień, w jakim można zidentyfkować podmiot, który podjął dane działania.

Autentyczność - stopień, w jakim można dowieść autentyczności podmiotu lub zasobu.

Łatwość utrzymania (ang. Maintainability)

Modułowość - stopień, w jakim system lub aplikacja jest stworzona z osobnych komponentów takich, że zmiana jednego ma minimalny wpływ na zmianę drugiego.

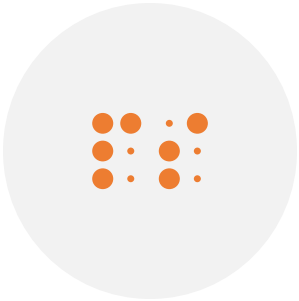
Łatwość ponownego użycia - stopień, w jakim część systemu może być wykorzystana w innym systemie lub do budowania innych systemów, modułów.

Łatwość analizy - stopień efektywności i wydajności z jakim istnieje możliwość oceny wpływu zmiany na produkt lub system lub diagnozy produktu w celu znalezienia przyczyny błędów lub części, które mają być zmienione.

Łatwość modyfikowania - stopień, w jakim system lub program mogą być efektywnie I wydajnie zmienione tak, aby nie wprowadzić innych błędów i nie pogorszyć jakości produktu.

Łatwość testowania - stopień, w jakim efektywne i wydajne kryteria testów mogą być stworzone i testy mogą być wykonywane.

Przenośność (ang. Portability)



ŁATWOŚĆ ADAPTACJI - STOPIEŃ, W JAKIM PRODUKT LUB SYSTEM MOŻE BYĆ EFEKTYWNIE I WYDAJNIE ZAADAPTOWANY DO INNYCH LUB ZMIENIAJĄCYCH SIĘ WARUNKÓW - SPRZĘT, OPROGRAMOWANIE, WYPOSAŻENIE, INNE ELEMENTY ŚRODOWISKA/KONTEKSTU.

ŁATWOŚĆ INSTALACJI - STOPIEŃ, W JAKIM EFEKTYWNIE I WYDAJNIE PRODUKT LUB SYSTEM MOŻE BYĆ Z SUKCESEM ZAINSTALOWANY LUB ODINSTALOWANY W DANYM ŚRODOWISKU.

ŁATWOŚĆ ZAMIANY - STOPIEŃ, W JAKIM PRODUKT MOŻE BYĆ ZASTĄPIONY INNYM PRODUKTEM DLA REALIZACJI TEGO SAMEGO CELU W DANYM ŚRODOWISKU.

Jakość użycia

EFEKTYWNOŚĆ (ANG. EFFECTIVENESS) - DOKŁADNOŚĆ I KOMPLETNOŚĆ, Z JAKĄ UŻYTKOWNICY OSIĄGAJĄ OKRESŁONE CELE.

SPRAWNOŚĆ (ANG. EFFICIENCY) - ILOŚĆ ZASOBÓW POTRZEBNA DO TEGO, ABY UŻYTKOWNICY MOGLI OSIĄGNĄĆ SWOJE CELE Z OKREŚLONĄ DOKŁADNOŚCIĄ I KOMPLETNOŚCIĄ.

SATYSFAKCJA (ANG. SATISFACTION) - STOPIEŃ SPEŁNIENIA POTRZEB UŻYTKOWNIKA ZWIĄZANYCH Z UŻYCIEM PRODUKTU LUB SYSTEMU W OKREŚLONYM KONTEKŚCIE UŻYCIA (UŻYTECZNOŚĆ, ZAUFANIE, PRZYJEMNOŚĆ, KOMFORT).

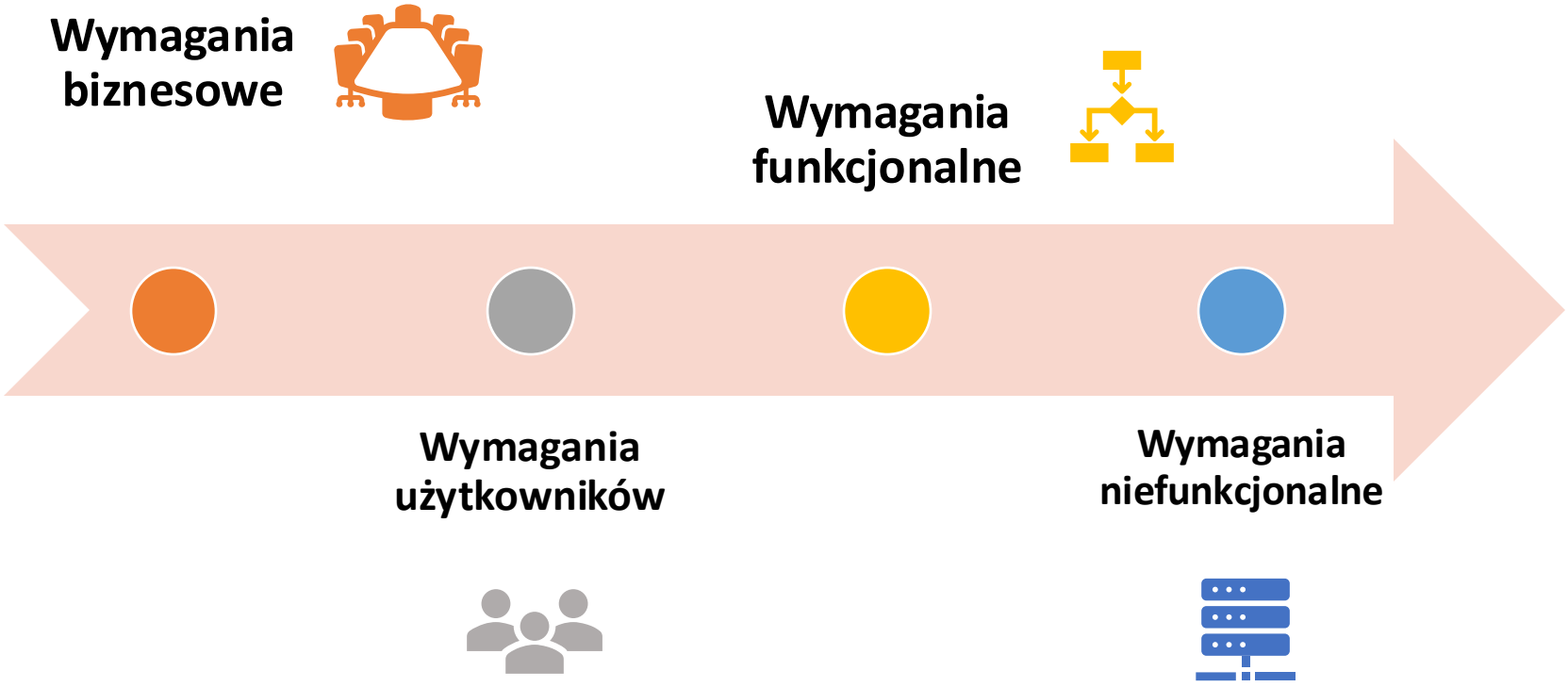
WOLNOŚĆ OD RYZYKA (ANG. FREEDOM FROM RISK) - ISTOTNE!
STOPIEŃ, W JAKIM PRODUKT LUB SYSTEM ŁAGODZI RYZYKO DOTYCZĄCE ASPEKTÓW EKONOMICZNYCH, ŻYCIA LUDZKIEGO, ZDROWIA LUB ŚRODOWISKA.

3. Gromadzenie wymagań

- Najmniej techniczna faza rozwoju systemu
- Wymaga umiejętności socjalnych, komunikacyjnych i menadżerskich
- Dostarcza (głownie tekstową) definicje wymagań

Hierarchia wymagań

Czyli od czego rozpocząć analizę?



Tradycyjne metody wydobywania wymagań

- Wywiady z klientami i ekspertami domeny biznesowej
- Kwestionariusze
- Obserwacja
- Studiowanie dokumentów i innych systemów klienta

Są proste i efektywne kosztowo, stosujemy je gdy są jasne cele i niskie ryzyko projektu.

Wywiady z klientami i ekspertami domeny

- **Z klientami** – głownie wymagania przypadków użycia
- **Z ekspertami domeny** – często prosty transfer wiedzy
- **Formalny wywiad strukturalny** (Structured interview)
  - Pytania otwarte (nieoczekiwane odpowiedzi)
  - Pytania zamknięte (lista możliwych odpowiedzi znana)
- **Nieformalny wywiad niestrukturalny**

Wywiad – rodzaje pytań

- **o szczegóły: pięć „W”: what, who, when, where, and why**
- **o wizje przyszłości**
- **o alternatywne idee -** to mogą być pytania osób uczestniczących w wywiadzie albo sugestie osób prowadzących wywiad
- **o minimalnie-akceptowalne rozwiązanie problem** dobre użyteczne systemy są systemami prostymi
- **o inne źródła informacji -** mogą odkryć ważne dokumenty i inne źródła wiedzy nieznane dla osoby prowadzącej wywiad
- **proszące o diagramy/modele** rysowane przez osobę uczestniczącą w wywiadzie w celu wyjaśnienia procesów biznesowych, mogą okazać się bezcenne dla zrozumienia wymagań

Czego unikać?

- **Pytania, których należy unikać**
  - **Pytania opiniotwórcze** (czy musimy to robić tak jak to robimy?)
  - **Pytania stronnicze** (nie zamierzasz chyba tak tego robić?)
  - **Pytania narzucające** (robisz to tak, czy nieprawda?)
- **Raport podsumowujący** wywiad powinien zostać wysłany do wszystkich osób uczestniczących w wywiadzie w ciągu jednego-dwóch dni z prośbą o komentarze

Kwestionariusze

- **Jako uzupełnienie** wywiadów
- Technika **pasywna**
  - zaleta – jest czas na rozważenie odpowiedzi
  - wada – nie ma możliwości klaryfikacji pytań i odpowiedzi
  - jakie grupy ludzi nie odpowiedziały? jak odpowiedziałyby?
- **Zamknięte pytania**
  - Pytania z lista **do wyboru** (multiple-choice) odpowiedzi - dodatkowe komentarze mogą być dozwolone
  - Pytania **o ocenę** (rating) (np. zgadzam się w pełni, zgadzam się, …) - kiedy szukamy opinii
  - Pytania **o rangę** (ranking) - przy pomocy liczb, procentów, itp.

Obserwacja

- Jako uzupełnienie wywiadów (a także kwestionariuszy)
- Wtedy kiedy użytkownik nie potrafi przekazać dostatecznej informacji i/lub ma fragmentaryczna wiedze
- Trzy formy:
  - **Pasywna**
    - bez przerywania pracy i bez bezpośrednich interwencji
    - kamera video może być użyta
  - **Aktywna**
  - **Wyjaśniająca**
    - wyjaśnianie tego co jest obserwowane
- Powinna być prowadzona przez dłuższy okres czasu, w rożnych okresach i przy rożnych obciążeniach praca
- Ludzie maja tendencje zachowywania się inaczej gdy są obserwowani

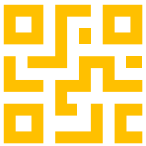
Studiowanie dokumentacji

- Zawsze stosowane, ale może dotyczyć wybranych części systemu
  - Dokumenty w organizacji
    - w tym procedury, reguły, opisy, plany, wykresy, wewnętrzna i zewnętrzna korespondencja
  - Formularze i raporty z istniejących systemów informatycznych
    - opisy wniosków o zmiany (change requests) - o naprawę wad i o wprowadzenie ulepszę
- Wymagania (z) **przypadków użycia**
- Wymagania (z) **wiedzy dziedzinowej**
  - czasopisma i książki z dziedziny
  - Systemy ERP (enterprise resource planning), CRM (customer relationship management) i SCM (supply chain management)
  - badania źródeł Internetowych

Nowoczesne metody wydobywania wymagań

- **Zawierają**: Prototypowanie, Burzę mózgów, Design Thinking

- Dają lepszy wgląd, ale przy wyższym koszcie i wysiłku
- Wtedy kiedy **ryzyko projektu** jest wysokie
  - niejasne cele,
  - nieudokumentowane procedury,
  - niestabilne wymagania,
  - wątpliwa ekspertyza użytkowników,
  - niedoświadczeni deweloperzy,
  - niedostateczne zainteresowanie użytkowników,



Prototypowanie

- Rozwiązanie **„szybkie i brudne”** (‘quick and dirty’) aby uzyskać „sprzężenie zwrotne” (feedback)
- Konieczne w projektach **złożonych** i **innowacyjnych**

- Dwa rodzaje prototypów:
  - Prototyp **do wyrzucenia** (throw-away prototype) - ukierunkowany tylko na wydobycie wymagań
  - Prototyp **ewolucyjny** - ukierunkowany na szybkość dostarczenia produktu

Burza mózgów – kiedy?

- aby sformułować nowe idee lub znaleźć rozwiązanie specyficznego problemu przez odsuniecie z punktu widzenia osobistych osądów, krytyki, reguł i zahamowań socjalnych
- aby osiągnąć zgodę/porozumienie miedzy zainteresowanymi stronami (stakeholders)
- zimna analiza i podejmowanie decyzji następuje później

Burza mózgów - proces

- przed burza mózgów, **prowadzący** warsztat (facilitator) dostarcza **opis problemu**
  - ogólne **pytania celowe** jako wyzwanie dla uczestników (np. jakie cechy charakterystyczne powinien mieć system? jakie są główne „obiekty biznesowe”?)
- 12 do 20 uczestników siedzących **wokół stołu** mających uczucie, że są „jedną osobą w tłumie” osób o równych prawach
- odpowiedzi do pytań celowych są **zapisywane i wysyłane** wokół stołu żeby stymulować więcej odpowiedzi i idei
- odpowiedzi i idee są **anonimowe**
- odpowiedzi i idee są **dyskutowane**
- **glosowanie** w celu definicji priorytetów odpowiedzi i idei

Design Thinking

Design Thinking to metoda tworzenia innowacyjnych produktów i usług w oparciu o głębokie zrozumienie problemów i potrzeb użytkowników.

ZAŁOŻENIA:

- **Koncentracja na użytkowniku** – zrozumienie jego uświadomionych i nieuświadomionych potrzeb
- **Interdyscyplinarny zespół** – spojrzenie na problem z wielu perspektyw
- **Eksperymentowanie** i częste testowanie hipotez – budowanie prototypów i zbieranie feedbacku od użytkowników
- W efekcie powstają rozwiązania, które są:
    - Pożądane przez użytkowników
    - Technologicznie wykonalne
    - Ekonomicznie uzasadnione

4. Dokumentowanie wymagań

- Wizja rozwiązania systemowego
- User stories
- Przypadki użycia

Definiowanie wizji rozwiązania

- Wizja rozwiązania jest podejściem zorientowanym na wartość biznesową dostarczanej usługi informatycznej (tzn. nie tylko systemu informatycznego)
  - żeby rozwiązać obecny (As-Is) problem biznesowy
  - żeby wesprzeć innowacje biznesowa (To-Be)

- Wizja rozwiązania ustanawia bliska więź miedzy biznesem a innymi zainteresowanymi stronami oraz integruje metody strategii biznesowej z możliwościami rozwiązania programistycznego

- Wizja rozwiązania powinna być zrealizowana wg zasady trzech „E”:
  - Wydajność (efficiency)
  - Skuteczność (effectiveness)
  - W zdefiniowanych ramach określających zakres rozwiązania (edge)

Trzy fazy procesu ustalania wizji rozwiązania

1) Eksploracja możliwości biznesowych
(Business capability exploration)

- określa możliwości biznesowe związane z tym jak rozwiązanie informatyczne może dostarczyć konkretnych rezultatów
- Ta faza opisuje możliwości (capability cases) – idee rozwiązań przedstawiające argumentacje biznesowa (business case) dla każdej możliwości

2) Wizja możliwości rozwiązania
(Solution capability envisioning)

- ma na celu rozwój wizji rozwiązania w celu zbudowania koncepcji rozwiązania (solution concept)
- Koncepcja rozwiązania uwzględnia kontekst biznesowy i tworzy przyszłe scenariusze nowych sposobów działania firmy
- Koncepcja rozwiązania precyzuje ostateczną architekturę rozwiązania

3) Projekt możliwości oprogramowania
(Software capability design)

- Decyduje o technologiach implementacji systemu, rozwija architekturę możliwości oprogramowania i opracowuje plany projektu i analizę ryzyka dla przypadku biznesowego
- Projekt możliwości oprogramowania jest działalnością w sferze modelowania oprogramowania.

Strategia implementacji i architektura możliwości

- Rozwój własny (custom development)
  - wykonywany wewnątrz organizacji i/lub
  - subkontraktowany do firm konsultingowych i deweloperskich
- Rozwój bazowany na pakietach (package-based development), który osiąga rozwiązanie w drodze kustomizacji istniejących pakietów oprogramowania, takich jak systemy ERP lub CRM.
- Rozwój bazowany na komponentach (component-based development), który osiąga rozwiązanie przez integrowanie komponentów oprogramowania od wielu dostawców i partnerów biznesowych, typowo z wykorzystaniem
  - SOA (**Service-Oriented Architecture**, **SOA**) – koncepcja tworzenia systemów informatycznych, w której główny nacisk stawia się na definiowanie usług, które spełnią wymagania użytkownika.
  - i/lub MDA **Model Driven Architecture** – określenie z dziedziny inżynierii oprogramowania, określające zbiór metod porządkujących proces tworzenia systemów komputerowych opartych na budowie modeli i ich transformacji.

Wymagania użytkowników

- **Historyjka użytkownika** to nieformalne, ogólne objaśnienie funkcji oprogramowania napisane z perspektywy użytkownika końcowego lub klienta (wg Atlassian).

„Jako [persona] [chcę], [aby]”

Jako pracownik chcę mieć możliwość utworzenia oferty, aby móc ją przekazać klientowi w krótkim czasie.

Jako kierownik chcę mieć możliwość tworzenia raportów sprzedaży, aby móc monitorować stan realizacji planów rocznych.

- Jednym z najbardziej popularnych sposobów definiowania wymagań użytkowników w metodyce Agile.

Struktura historyjki użytkownika

- WHO - „Jako [**persona**]”: dla kogo tworzymy nasz produkt?
- WHAT - „**Chcę**”: tutaj opisujemy jego zamiary, a nie funkcje, z których korzysta. Co próbuje tak naprawdę osiągnąć?
- WHY - „**Aby**”: w jaki sposób jego bezpośrednia chęć zrobienia czegoś wpisuje się w szerszą perspektywę? Jakie ogólne korzyści próbuje osiągnąć? Jak wygląda problem wymagający rozwiązania?

Kryteria akceptacji historyjki

- kryteria zorientowane na reguły (szablon listy kontrolnej **INVEST**)
  - **I** – **Independent** - niezależna od innych User Story
  - **N** – **Negotiable** - negocjowalna, powinna otwierać rozmowę nie ją zamykać, dopóki stanowi ona część Backlogu, dopóty może być modyfikowana, ukonkretniana
  - **V** – **Valuable** - wartościowa z perspektywy użytkownika, powinna oferować użytkownikowi jakąś wartość
  - **E** – **Estimable** - możliwa do oszacowania, tylko w takiej formule można ją podzielić na zadania, a więc zaplanować w czasie
  - **S** – **Small** - mała - jej realizacja powinna być możliwa do wykonania w możliwie najkrótszym czasie
  - **T** – **Testable** - testowalna, a więc powinna skutkować jednoznacznym rozstrzygnięciem - Skończona / Nie skończona.
- kryteria zorientowane na scenariusze
  - Biorąc pod uwagę pewne warunki wstępne
  - Kiedy wykonuję jakąś akcję
  - Wtedy oczekuję jakiegoś rezultatu.

Korzyście ze stosowania historyjek

- **Historyjki kładą nacisk na użytkownika.** Lista do wykonania sprawia, że zespół koncentruje się na zadaniach, które trzeba odhaczyć, natomiast zbiór historyjek skupia uwagę zespołu na rozwiązywaniu problemów prawdziwych użytkowników.

- **Historyjki sprzyjają współpracy.** Gdy ostateczny cel zostanie zdefiniowany, zespół może pracować razem, aby ustalić, w jaki sposób najlepiej zapewnić użytkownikowi korzyści i zrealizować cel.

- **Historyjki stymulują poszukiwanie twórczych rozwiązań.** Historyjki zachęcają zespół do krytycznego i kreatywnego myślenia o najlepszych sposobach osiągnięcia ostatecznego celu.

- **Historyjki nadają impet.** Każda kolejna historyjka pozwala cieszyć się zespołowi programistycznemu ze sprostania niewielkiemu wyzwaniu i odniesieniu małego zwycięstwa, co napędza jego dynamikę.

Źródło

5. Negocjacja i walidacja wymagań

- wydobyte wymagania musza być poddane uważnej negocjacji i walidacji z zainteresowanymi stronami

Negocjacja i walidacja wymagań

- Niezbędne ponieważ wymagania:
  - mogą **nakładać się** i **być w konflikcie** matryca zależności wymagań (requirements dependency matrix) (następny slajd)
  - mogą być **dwuznaczne** i **nierealistyczne**
  - mogą być **niewydobyte**
  - mogą być **poza zakresem** (naznaczonym przez model referencyjny, taki jak diagram kontekstu (wyjaśniony dalej))
    - czasem poza zakresem projektu, ale w zakresie systemu informacyjnego (wymagania mogą być zbyt trudne do implementacji komputerowej, mogą mieć niski priorytet, mogą być zaimplementowane w hardwarze)
- faza często wykonywana równolegle z wydobywaniem wymagań
- faza nierozłączna od produkcji dokumentacji wymagań
  - negocjacje zaczynają się od pierwszego szkicu dokumentu wymagań
  - walidacja ocenia i przypieczętowuje dokument

Matryca zależności wymagań

Wymaganie	***R1***	***R2***	***R3***	***R4***
R1				
R2	*Konflikt*			
R3				
R4		*Nałożenie sie*	*Nałożenie sie*	

Ryzyko i priorytety wymagań

- **Ryzyko** jest zagrożeniem dla planu projektu
- Ryzyko określa **wykonalność** (feasibility) projektu
- **Analiza ryzyka** identyfikuje wymagania, które maja największą szansę zagrozić projektowi
- **Definiowanie priorytetów** jest konieczna żeby ułatwiać zmiany zakresu projektu gdy są opóźnienia
- **Kategorie ryzyka**
  - Techniczne
  - Wydajność
  - Bezpieczeństwo
  - Integralność bazy danych
  - Proces rozwoju
  - Polityczne
  - Prawne
  - Zmienność wymagań

Bez względu na rodzaj przyjętej metodyki pracy analiza biznesowa wymaga

- zadawania pytań,
- wyciągania wniosków,
- przetwarzania,
- analizowania,
- optymalizowania.

Najczęstsze błędy w analizie biznesowej

1. Błędnie określona potrzeba biznesowa
2. Stosowanie nieoptymalnych rozwiązań biznesowych (dług technologiczny)
3. Przedstawianie gotowych rozwiązań
4. Bezkrytyczne zbieranie wymagań
5. Niezrozumienie dziedziny
6. Brak konsultacji
7. Brak priorytetów

Dziękuję za uwagę