

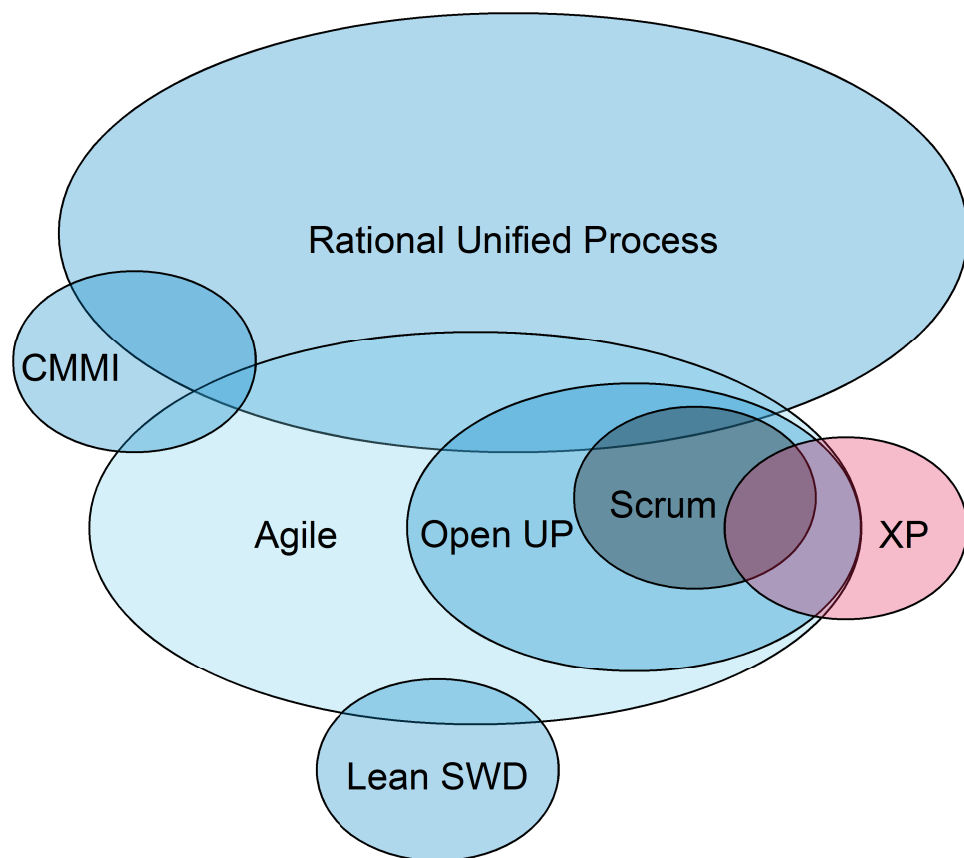
Jak rozwiązać nierozwiązaln e problemy w projektach ...i mieć lepsze życie

Senior Consultant
Tieto,
marcin.kokott@tieto.com

Czy ja mam jakiś problem?

- Nie masz problemu jeżeli twój produkt jest perfekcyjny...
- Perfekcyjny dla ludzi:
 - Używających twój produkt
 - Inwestujących w twój produkt by rozwiązać swoje problemy za które są odpowiedzialni
 - Sprzedających, instalujących, wspierających i utrzymujących go
 - Projektujących, implementujących, weryfikujących, integrujących i utrzymujących twój produkt przez długi czas
- Jeżeli twój produkt zadowala wszystkich tych ludzi – **nie masz problemu!**

Zobaczyć nie znaczy naprawić



Kim jestem

- Programista, lider zespołu, kierownik projektu
- ...również komik, ratownik medyczny
- ...ponad 8 lat doświadczenia w IT
- ...zajmował się implementacją Agile & Lean w różnego typu projektach przez ostatnie 2 lata. Najczęściej środowiska rozproszone geograficznie (Skandynawia, Niemcy, Holandia, Polska, Indie czy Chiny)
- Aktualnie Agile/Lean Coach wspierający krytyczne biznesowo projekty w Tieto
- ...tworzył i prowadził treningi i warsztaty z tematyki Agile i Lean dla prawie wszystkich szczebli w organizacji jak również i innych firm.



To co z tymi problemami?

- Wyobraźmy sobie sytuację:
 - Jesteś programistą, dostajesz zadanie do oszacowania
 - ...z funkcjonalności, której jeszcze nie wytworzyliście...
 - ...z obszaru projektu z której nie czujesz się za dobrze...
 - ...oszacowania mają być za godzinę...
 - ...i klient mówi, że powinny być mniejsze n
- Czy rozwiązanie jest:
 - Win-win ?
 - Trwałe?
 - Długoterminowe?
- Przywykliśmy do faktu by reagować na symptomy...



Wszyscy jesteśmy ludźmi

- Percepcyjna ślepotą
- Błędy w argumentacji
 - Grupa A: „Kara śmierci ma sens”
 - Grupa B: „Przeciwno”
 - Dwa fikcyjne analizy naukowe: jedna popierająca (1), druga przeciwko (2)
 - Grupa A: ignoruje (akceptuje) nieścisłości w analizie (1), bardzo dokładnie zauważa wszystkie nieścisłości w analizie (2)
 - Grupa B: dokładnie odwrotnie...
- Zawsze kwestionuj sytuacje z jednym punktem postrzegania rzeczywistości



Bądź jak dziecko



- Najczęściej wraz z doświadczeniem otrzymujemy również:
 - Postrzeganie tunelowe:
 - „Moim zadaniem jest tylko zrobienie tej części i pójście do domu”
 - Ograniczenia w naszych głowach
 - „Wszyscy naokoło tak robią, więc tego się nie da zmienić”
 - Założenia
 - Programista: „Kierownictwo nigdy się na to nie zgodzi”
 - Kierownictwoi: „Świetna idea, ale klient nigdy na to nie pójdzie”
 - Klient: „Lubię to rozwiązanie ale obawiam się, że zespół sobie z tym nie poradzi””
 - Przyzwyczajenia
 - „Zawsze tak to robimy”
 - Strach

Zobaczmy słonia



Problemy lubią granice

- Granice pomiędzy grupami/oddziałami
 - Programiści & Testerzy
 - Klient & firma IT
 - Kierownictwo & Zespoły
- Kolejki i sterty „to-do”
 - Sprzedaż & Development
 - Development & Integracja
 - Development & Testowanie
- Fizyczna odległość
 - Projekt w różnych krajach
 - ...a czasem nawet kilka metrów



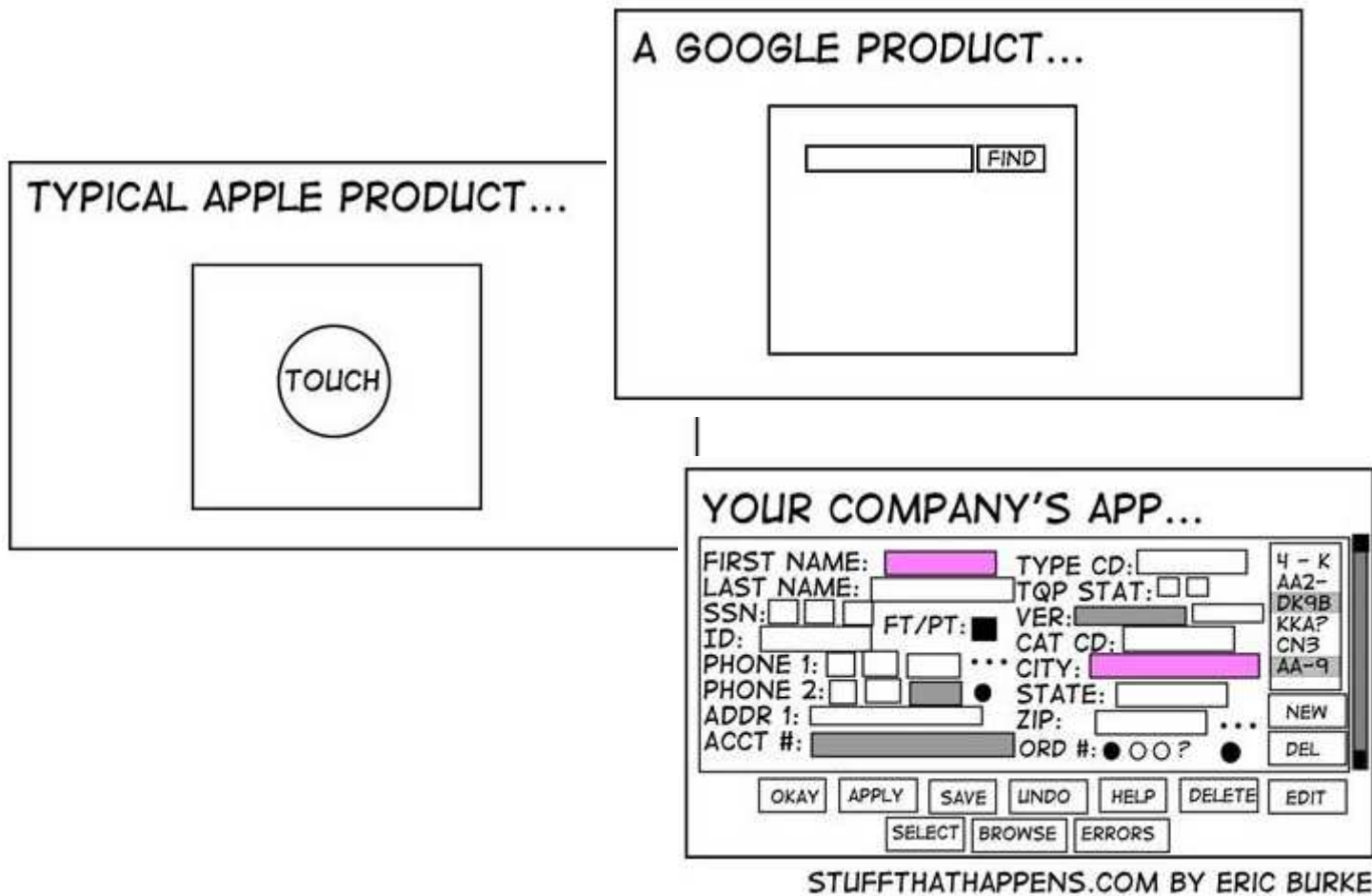
Magia



- Celem jest dostarczenie wartości klientowi
- Zaczniemy widzieć przez perspektywę klienta
- System:
 - „W pełni kompletny i przydatny produkt wraz ze wszystkim co jest potrzebne do wytworzenia i rozdystrybuowania go”
- **Klient kupuje i używa SYSTEMU**

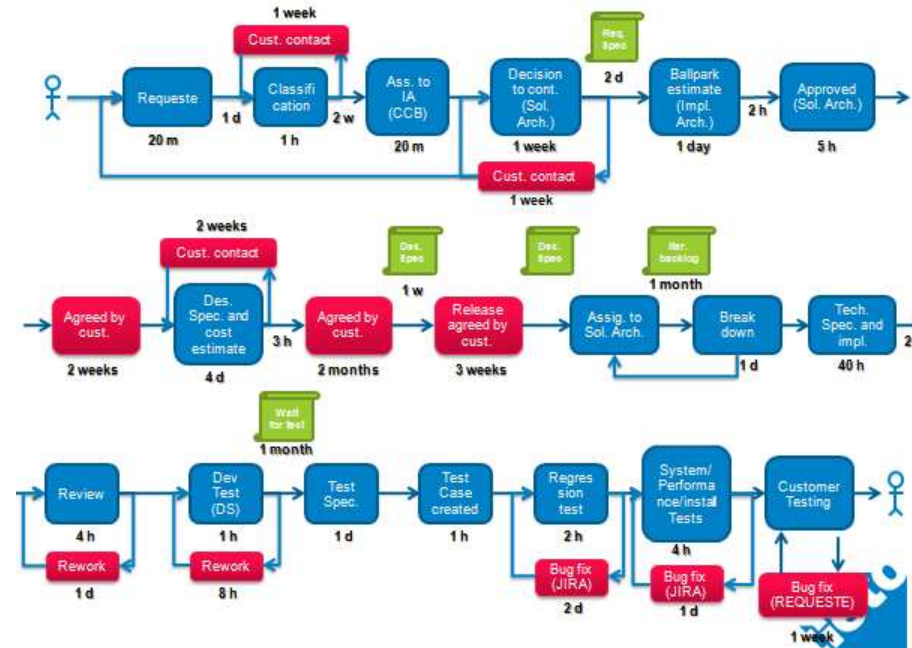


Chodzi o wartość



„Problem komunikacyjny”

- Scrum Master
 - „Mamy problem z komunika
- Product Owner:
 - „Nasz development nie jest
- Development Manager
 - „Implementacja trwa za długo”
- Sales
 - „Nie dotrzymujemy naszych obietnic”
- CTO
 - „Mamy za długi czas wejścia na rynek”
 - (Time-to-Market)



„Już wiele razy próbowaliśmy rozwiązać ten problem... Każdy dział wymyślał rozwiązania”

No to co?... I dlaczego?

WIZJA

**Rzeczywistość
(aktualna
sytuacja)**

**Mały krok
(Kaizen)**

**Postęp
(zmierzalny)**

**Cel
długoterminowy**

Background

- Internal market analysis - Revenue & Profit
- Development teams - Few developers, slow to get
- Overhead costs - Time to develop games mostly increasing due to declining technical quality
- Revenue in 2009 (€M)

Current Condition

Goal / Target Condition

- 4x faster cycle time
- 2x fewer required authors
- 200% improvement in revenue

Root Cause Analysis

Owner: Tieto
Method: Kaizen
Date: 18 May 2009

Countmeasures:

1. Cross Functional Teams - Simplifies design through deployment
 - o Fewer dependencies - save spend 75% of time waiting/engendering
2. Abolish all but most pertinent 3 games in each space. 2x OME games per cross functional team at a time
 - o 2x faster delivery from reduced task switching
3. Engage developers in playing games and selecting ideas
 - o Easier to get system will cut 1.2 years from schedule
 - o 200% more profits to just with best competitor
 - o Improved training on what games to develop
 - o More fun games, more popular

Confirmation (Results)

1. Cross Functional Teams
 - o Half as much time waiting
2. One game at a time
 - o Quicker a finished, time to complete game is 4 months (6x)
 - o Reduced OME development - Reduced OME from 12x to 2x for
3. Engage developers in playing games and selecting ideas
 - o One team taking time to play is producing more innovative games.
 - o Impact on profits is TBD.

Follow-up

1. Consider more cross training of team members to reduce waiting for expertise
2. Reduce difficulty of integration and deployment steps
3. Improve processes for generating and selecting game ideas
 - a. Broaden talent if identifiable/available
 - b. Improve skills/process of best people already in company
 - c. Broaden both participation in selection and game playing experience of everyone in the company.
4. Continue improvement of fixed game components/engines to improve development throughput and reduce defects.

Narzędzia:

- „5 Why’s”
- Current Reality Tree
- Theory of Constraints
- Value stream map
- MBTI (typy osobowości)



Więc

- **Znajdźmy dziurę w łódce**
- **Zobaczmy całego słonia**
- **Zbudujmy katedrę**
- **Zacznijmy ze sobą rozmawiać**
- **Badźmy dziećmi...**
- **...i nigdy nie przestawajmy się zmieniać!**

