

Agile UX



Michał Trzaskowski

Co czyni interfejs użytkownika lepszym?
(bardziej używalnym)

Prosty i naturalny język

Minimalne wymagania pamięciowe
na użytkownika

Użyteczny feedback aplikacji

Wyjścia

Droga na skróty

Ochrona przed błędami

Prostota i konsekwencja

„Zwinne zagrożenia”

Współdziałanie zainteresowanych

- Projektanci (Designers)
- Deweloperzy
- Product owner
- Ekspertci domenowi
- Użytkownicy

Projektant ? To ten gość od CSS-a ?

Współdziałanie zainteresowanych

- Projektanci (Designers)
- Deweloperzy
- Product owner
- Ekspertci domenowi
- Użytkownicy

Apetyczny „Product Owner”

Współdziałanie zainteresowanych

- Projektanci (Designers)
- Deweloperzy
- Product owner
- Ekspertci domenowi
- Użytkownicy

Uświęcone ustalenia i procedury

„fax case”

<http://davenicolette.wordpress.com/2012/08/04/my-personal-agile-journey/>

Współdziałanie zainteresowanych

- Projektanci (Designers)
- Deweloperzy
- Product owner
- Ekspertci domenowi
- Użytkownicy

Klient \neq Użytkownik

Użytkownik czy Użytkownicy ?

Customization (Personalizacja)

Zwinna specyfikacja wymagań

„canonical User Story format”

<http://davenicolette.wordpress.com/2012/08/04/my-personal-agile-journey/>

Wymagania jawne

vs

Wymagania niejawne

Wymagania oczywiste

vs

Wymagania nieoczywiste

„Zwinna wycena”

Affinity Estimating

„Zwinne technologie“

Proces projektowania (Designing process)



- Szkicowanie
- Szkielet
- Prototyp
- Detale

Iteracyjna

prostota i konsekwencja ?

Iteracje początkowe



Kolejne iteracje...



Iteracje końcowe

Szkic aplikacji

Prototyp

Feedback !

Decyzje technologiczne

Szkic

Szkielet

Feedback !

Implementacja + Detale

Refaktoring

Dodatkowe przejścia i wyjścia

Wyrównanie

Optymalizacja

Lokalizacja



Dziękuję za uwagę !



We are hiring!

:)