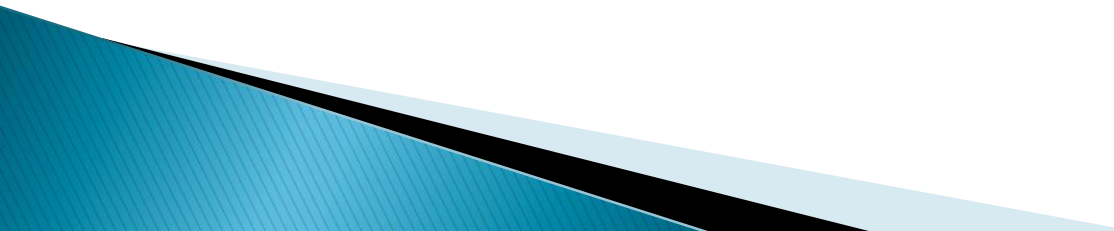



# Agility in Context

Streszczenie przygotowane przez  
Szymona Giżeckiego

# Agenda

- ▶ Programowanie zwinne – co to?
  - ▶ Kiedy stosować?
  - ▶ A co jeśli nie spełniamy wymaganych kryteriów? – experiment badający 40 firm IT
  - ▶ Podsumowanie
- 

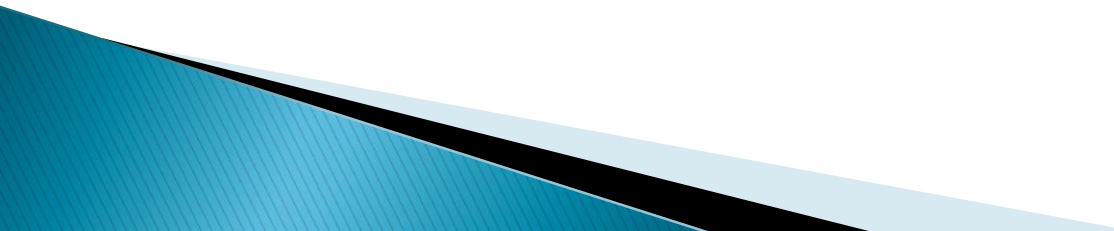
# Programowanie zwinne (Agile software development)

- ▶ Grupa metodyk wytwarzania oprogramowania oparta o programowanie iteracyjne
  - ▶ Częste, szybkie, stałe dostarczanie działającego kodu
  - ▶ Mało dokumentacji
  - ▶ Preferowany bezpośredni kontakt (wewnątrz teamów jak i z klientem)
  - ▶ Samozarządzalność zespołów – programiści sami decydują jak osiągnąć postawione cele
- 

# Programowanie zwinne

- ▶ Najpopularniejsze metodyki to:
  - SCRUM
  - XP
- ▶ W opisie tych metodyk częstymi zwrotami są: „nie może”, „musi”, itd.
- ▶ Cytując ewangelistów:
  - Projekty, które nie adaptują wszystkich praktyk z manuala nazywane są „Scrum-butts”
  - Testy „the Ten Ways You are Not AGILE”
  - „If you are almost doing XP, then you are not doing XP”
- ▶ Test: <http://www.scrumbutt.me/>

# Idealne zastosowanie

- ▶ Nie duże drużyny i projekty
  - ▶ Reprezentanci klienta na miejscu lub też o bardzo wysokiej dostępności
  - ▶ Zkoncentrowanie na implementacji
  - ▶ Wczesne testowanie
  - ▶ Częsty feedback
- 

# Zakres eksperymentu

P#	Positions	Method	Org. Size	Location	Domain	Team Size	Project (months)	Iteration (weeks)
P1-P7	Dev X 3, BA, AC, Tester, Cust Rep	Scrum	M	NZ	Health	7	9	2
P8	Cust Rep	Scrum & XP	L	NZ	Social Services	4 to 10	3 to 12	2
P9-P15	Dev X 5, AC, SM	Scrum & XP	S	NZ	Environment	4 to 6	12	1
P16	SM	Scrum & XP	S	NZ	E-commerce	4	2	4
P17	AC	Scrum & XP	XL	NZ	Transportation & Telecom	6 to 15	12	4
P18	Cust Rep	Scrum	XS	NZ	Entertainment	6 to 8	9	4
P19	AC	Scrum & XP	S	NZ	Government Education	4 to 9	4	2
P20	AC	Scrum & XP	XS	NZ	Software Development	8	12	1

# Problemy i rozwiązania

Contexts	Adaptation Strategies
Lack of Customer Involvement	Story Owners Customer Proxy Simulation
Fixed-bid Contracts	Providing Options Buffering
Design/Architecture Intensive	Information Architecture Design Pipeline Walking Skeleton
Documentation Intensive	Project Dictionary Comprehensive Docs.
Slow Rate of Change	Working from Requirements

# Brak zaangażowania klientów

- ▶ Jedno z proponowanych rozwiązań:  
„Właściciele historii/pięter” (*ang. Story owner*)
- ▶ Jedno piętro: mniej niż tydzień
- ▶ Osiągnięto:
  - Nie było osoby, która musiałaby być cały czas dostępna
  - Łatwiejszą synchronizacją programistów z klientem
  - Osoby odpowiedzialne za piętra „utożsamiały” się z nimi



# Mało zmian

- ▶ Wątpliwości czy użyć „programowania zwinnego”
- ▶ Tworzenie historii z wymagań i postępowanie jak w standardowym przypadku
- ▶ *“I’m not aware of a lot of changes that have been requested... But like we’ve sort of come across this situation where we’ve gone ‘hey, this would work much better if we do it this way’. And we’ve taken it back and maybe demoed what the screens would look like instead and got approval.”—P13, Senior Developer*

# Wnioski

- ▶ Menedżerzy i programiści byli bardzo zadowoleni z adoptowania tylko wybranych praktyk „programowania zwinnego”
  - ▶ Wszyscy deklarowali, że nie chcą rezygnować z tej metodologii
  - ▶ Podział na praktyki zależne od kontekstu i niezależne
- 