

Acceptance Test Driven Development wspierane przez narzędzie ROBOT Framework

Edyta Tomalik

Grzegorz Ziemiecki

for a
world
in motion™

Tradycyjne podejście



Wyzwania

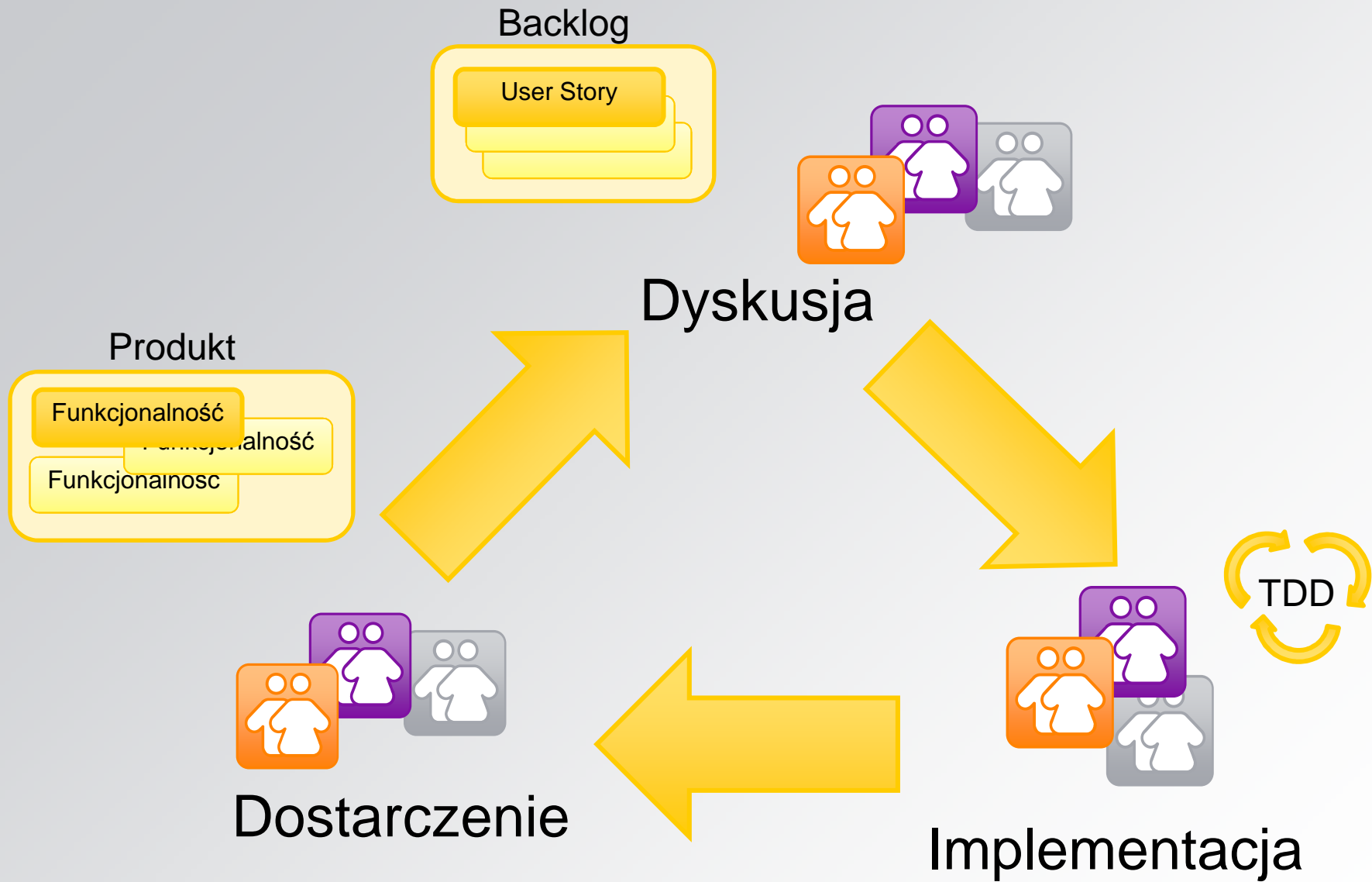
- Dynamiczne zmiany GUI
- Brak wystarczająco stabilnej wersji oprogramowania
- Brak wystarczającej ilości czasu na projektowanie przypadków testowych
- Ograniczenia budżetowe
- Niedojrzała metodologia automatyzacji testów
- Duże skomplikowanie kodu

Acceptance Test Driven Development

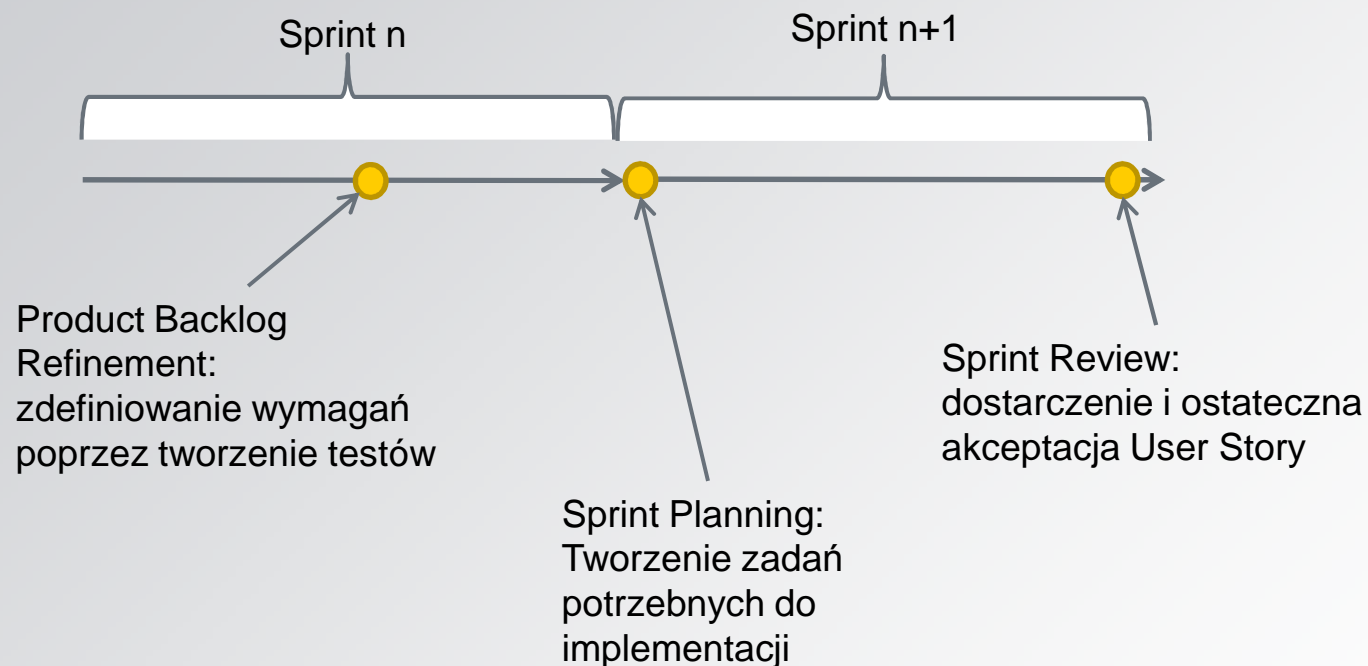
- Zróżnicowana terminologia
- Wspólne zrozumienie wymagań
- Warsztaty sposobem na zdefiniowanie wymagań
- Wymagania są wykonywalne
- Testy jako wymagania, wymagania jako testy
- Równoczesna implementacja i testowanie
- Zapobieganie zamiast detekcji
- „Żyjąca” dokumentacja



Cykl ATDD

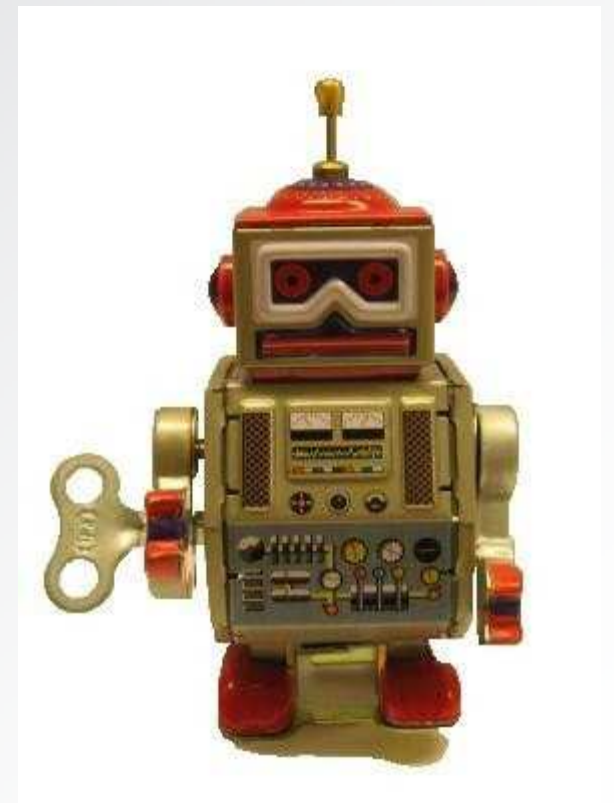


Mapowanie ATDD do procesu Scrum'owego



Dlaczego ROBOT Framework?

- Oparty o wysokopoziomowe słowa kluczowe
- Dopasowany do ATDD ale nie tylko
- Dokumentacja zintegrowana z testami
- Możliwość „tagowania” testów
- Niezależny od platformy
- Modułarna architektura
- Liczne biblioteki
- RIDE jako intuicyjny edytor testów
- Open Source



Kilka słów o przykładowej aplikacji



- Aplikacja do monitorowania sieci telekomunikacyjnej
- Alarmy wysyłane z elementów sieci do systemu zarządzającego
- Monitorowanie dostępne z poziomu linii poleceń
- Rozszerzamy funkcjonalność o możliwość zatwierdzania oraz usuwania alarmów lokalnie poprzez GUI



US: Jako administrator chcę mieć możliwość zatwierdzania oraz usuwania alarmów z poziomu GUI

Dyskusja

- Określenie kryteriów akceptacji
- Stawianie pytań
 - „Jak wyświetlany jest alarm?”
 - „Czy zmiana statusu alarmu jest możliwa tylko dla administratora?”
 - „Jakie prawa mają inni użytkownicy?”
 - „Czy jest możliwość przywrócenia usuniętego alarmu?”
 - „Czy dostępna jest historia alarmu?”
- Proste wymaganie komplikuje się
- Opisujemy CO? a nie JAK?
- Projektowanie testów
 - „Tylko administrator może zmieniać status alarmu”
 - „Alarm może zostać usunięty tylko w przypadku kiedy został wcześniej potwierdzony”
- Testy są „czerwone”



Wdzielenie przypadków testowych

- Przeniesienie przypadków testowych do narzędzia automatyzującego testy

„Tylko administrator może zmieniać status alarmu”

ALARMS

Setting	Value
Documentation	Test suite checks the access privileges for different users. Only administrator is allowed to change the alarms' status. Default user should not be able to change alarms' status

Test Case	Action	Arguments
Administrator is able to modify status of alarms	[Documentation]	This test case checks the ability of administrator account to modify the status of alarms.
	Start Application	
	Log In As Administrator	
	Open Window	Alarms
	Select Alarm	ID
	Modify Status	Acknowledge
	Check Status	ID
	Status Should Be	'acknowledged'
	Logout	

Implementacja oraz dostarczenie

- Celem jest test zakończony sukcesem
- Testy uruchamiane cyklicznie podczas fazy implementacji
- Implementacja przy użyciu metodyki TDD
- Testy eksploracyjne – ujawnienie ukrytych wymagań
- Demonstracja funkcjonalności podczas sprint review
- Zwiększenie zrozumienia

Dobre praktyki

- Testowanie pojedynczych warunków
- Konsekwentna organizacja testów
- Izolacja przypadków testowych
- Krótkie przypadki testowe
- Proste nazewnictwo słów kluczowych
- Dokumentacja jest bardzo istotna

Spróbuj

- Dyskutować nad wymaganiami podczas warsztatów
- Wyjaśniać zanim napiszesz test
- Operować konkretnymi przykładami
- Uruchamiać testy z dużą częstotliwością (CI)
- Narzędzi dopasowanych do ATDD
- Product Owner tworzy testy?

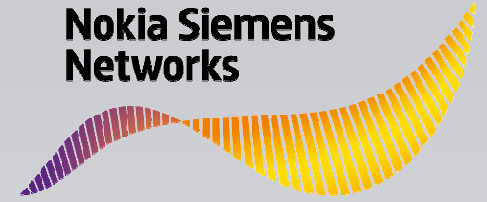
Unikaj

- Myślenia, że ATDD jest tylko dla testerów
- Mylenia TDD z ATDD
- Definiowania wymagań tylko przez analityków
- Dublowania opisów wymagań
- Powielania testów
- Skomplikowanych narzędzi dla testów automatycznych
- Komputerów i projektorów podczas warsztatów

Źródła

1. Adzic Gojko „Specification By Example”
2. Larman Craig, Vodde Bas „Practices for Scaling Lean & Agile Development”
3. www.robotframework.org

Nokia Siemens
Networks



for a
world
in motion™