

Liczba ankietowanych: 49

Ankieta została opublikowana na Polish Agile User Group, pl.comp.java i pl.comp.programming

1. Stand-up/ SCRUM

Codziennie robimy stand-up/SCRUM.	27	55%
Stand-up jest krótki i zawsze o tej samej porze.	28	57%
Klient uczestniczy w stand-up.	9	18%

2. Iteracje i planowanie

	Response Count	
Iteracje są ustalonej długości.	34	69%
Nigdy nie zmieniamy terminu iteracji.	24	49%
Wymagania są zamrożone po rozpoczęciu iteracji.	16	33%
Każda iteracja rozpoczyna się spotkaniem, na którym planujemy zadania do wykonania.	35	71%
Klient jest obecny przy planowaniu iteracji.	18	37%
Planujemy iteracje przy użyciu "user stories".	18	37%
Szacujemy i notujemy planowany czas wykonania każdego zadania.	32	65%
Zadania mają "acceptance criteria".	14	29%

3. Jak długa jest iteracja?

	Response Count	Response Percent
1 tydzień	6	15%
2 tygodnie	16	41%
3 tygodnie	5	13%
4 tygodnie	7	18%
Inna	5	13%

4. Zarządzanie iteracją

	Response Count	
Używamy "Index cards" i "Status Board".	16	33%
Mierzymy wydajność (Velocity).	16	33%
Używamy listy zadań (Task Backlog).	32	65%
Używamy systemu do rejestracji błędów.	34	69%

5. Praca nad kodem

	Response Count	
Pracujemy w parach (pair-programming) więcej niż 3 godziny dziennie, codziennie	6	12%
Każdy programista jest w stanie zmodyfikować dowolny moduł systemu bez obawy o wprowadzenie błędów do systemu.	17	35%

Wszyscy programiści w zespole stosują refactoring.	28	57%
Regularnie omawiamy co można zrobić by poprawić pracę zespołu.	29	59%
Robimy regularnie "Code reviews"	22	45%
Piszemy niewiele dokumentacji, nasz kod dokumentuje się sam.	24	49%
Używamy czasem narzędzi do mierzenia jakości kodu (test coverage, duplicates, inspections)	25	51%
Narzędzia do mierzenia jakości kodu są zintegrowane z "continuous integration".	11	22%
Robimy check-in wszystkich zmian przynajmniej raz dziennie.	27	55%
Check-iny są małe - średnio 5 plików.	22	45%
Możemy rozmawiać o kodzie napisanym przez innych programistów bez obawy, że ci poczują się zagrożeni/urazeni.	31	63%

6. Unit-testy

	Response Count	
Używamy Test Driven Development	18	37%
Używamy Test First Development	5	10%
Każdy może uruchomić unit-testy na swojej maszynie bez dodatkowej konfiguracji.	24	49%
Unit-testy trwają mniej niż 60 sekund.	9	18%

7. Continuous Integration

	Response Count	
Mamy tzw. "one-click" build	22	45%
Używamy "continuous integration" (przy każdym check-in do repozytorium).	15	31%
Wszyscy wiemy od razu jeśli build jest zepsuty.	21	43%
Naprawiamy build jeśli jest zepsuty zanim wolno zrobić "check-in".	13	27%
Hot-fix releases tylko przy użyciu "hot-fix branch".	5	10%

