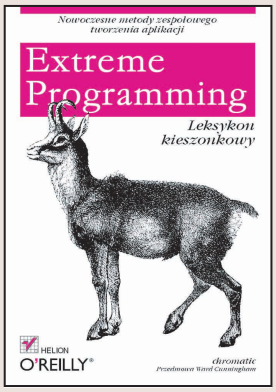
Extreme Programming. Leksykon kieszonkowy



Autor: chromatic
Tłumaczenie: Rafał Jońca
ISBN: 83-7361-343-9
Tytuł oryginału: Extreme Programming Pocket Guide
Format: B5, stron: 104

Wydajne programowanie (ang. Extreme Programming, XP) to nowe podejście do tworzenia oprogramowania oparte o najlepsze z technik ze szczególnym uwzględnieniem prostoty i pracy zespołowej. XP to ciągłe testowanie i przeglądanie kodu oraz znaczne zaangażowanie klienta w proces tworzenia aplikacji. Choć metody te przemawiają do programistów, ich praktykowanie wymaga cierpliwości i jest nie lada wyzwaniem.

Niniejsza książka wyjaśnia podstawowe techniki w ujęciu całościowym. Niezależnie od tego, czy jesteś programistą, klientem lub menedżerem; czy zaczynasz projekt od początku lub chcesz poprawić już istniejący, Extreme Programming z pewnością może Cię wiele nauczyć. Warto mieć satysfakcję z pisania oprogramowania.

W leksykonie omówiono:

- Techniki XP
- Elementy XP
- Zdarzenia w XP
- Kodowanie w stylu XP
- Wdrożenie XP

Zamiast zagłębiać się w setkach stron na temat Extreme Programming, sprawdź jak wiele informacji udało się zmieścić w tej małej książeczce. Będzie Ci ona towarzyszyć i służyć pomocą w całym procesie tworzenia aplikacji.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Rozdział 5. Role w XP

Każdy projekt XP zawiera kilka różnych ról — każda z nich ma własne prawa i zakres odpowiedzialności. XP stara się polepszyć komunikację między klientem a programistą, dzięki wyraźnemu rozdzieleniu zadań stojących przed tymi dwoma osobami. Jeśli zadanie ma zostać wykonane właściwe, trzeba rozmawiać z drugą grupą.

XP daje programistom autorytet podejmowania decyzji technicznych, gdyż dobrze się na nich znają. Klient ma za zadanie podejmować decyzje biznesowe. Obydwie dziedziny wpływają na siebie. Dokładne rozdzielenie zadań zwiększa szansę odniesienia sukcesu.

Klient

Klient steruje projektem. Definiuje go i określa jego cele. Im dokładniejsza jest jego praca i częstszy udział w zebraniach, tym większe prawdopodobieństwo odniesienia sukcesu.

Klient podejmuje decyzje biznesowe. Odpowiada za nie, gdyż jest obeznany z tym tematem. Jego celem jest określenie celów projektu (funkcji programu). Musi odpowiedzieć na pytania: „Co powinna robić dana funkcja?", „W jaki sposób poznamy, iż jest gotowa?", „Ile czasu możemy nad nią spędzić?", „Kiedy należy ją zaimplementować?".

Klient współpracuje z programistami. Pisze karty funkcji, wyjaśniające zasadę działania funkcji, i tworzy harmonogram. Odpowiada na pytanie *co?*. Uczestniczy w planowaniu i wybiera funkcje wprowadzane w następnej iteracji. Odpowiada na pytania *kiedy?* i *ile?*. Tworzy i wykonuje teksty akceptacyjne (przy pomocy programisty), aby sprawdzić, czy funkcja jest już gotowa. Odpowiada na pytanie *czy gotowe?*.

Klient reprezentuje użytkownika końcowego. W projekcie prowadzonym wewnątrz funkcji może nawet być końcowym użytkownikiem. W innych sytuacjach służy jako pośrednik. Odpowiada za identyfikację potrzeb z punktu widzenia użytkownika. Programiści troszczą się o kwestie techniczne.

Klient odpowiada też za stronę finansową projektu. Dąży do maksymalizacji zysku. W dowolnym momencie oprogramowanie powinno zawierać najbardziej istotne funkcje, które zostały dodane do harmonogramu zgodnie z posiadaną wiedzą.

Klient w technikach XP korzysta z kilku źródeł informacji. Musi rozumieć zagadnienia biznesowe, dotyczące projektu, nawet jeśli te zmieniają się wraz z upływem czasu. Czy funkcja jest teraz mniej lub bardziej ważna niż wtedy, gdy była definiowana? Czy funkcja może zostać opóźniona, zaniechana lub uproszczona? Klient musi być w stanie w dowolnym momencie ocenić projekt. Które funkcje są gotowe? W jakim stopniu spełniają one założenia? Z drugiej strony klient musi znać techniczne zależności, przede wszystkim ich wpływ na wartość i ryzyko wprowadzenia funkcji. Czy lepiej dodać do harmonogramu własną funkcję, czy może skorzystać z propozycji programisty?

XP zawsze traktuje klienta jako jedną osobę. Jeśli jest tylko pośrednikiem z klientami lub przedstawicielem inwestora, musi mówić jednym głosem. Przyjmuje rolę autorytetu, który ponosi odpowiedzialność za swoje decyzje.

Prawa klienta

XP określa kilka praw klienta.

- *Maksymalizacja zysków*, aby w harmonogramie znalazły się zawsze najpotrzebniejsze funkcje (patrz „2. technika biznesowa: zabawa w planowanie" z rozdziału 2.).

- *Dostosowanie możliwości projektu do zmian harmonogramu*. Dobieranie funkcji do dodania lub usunięcia z iteracji, jeśli przybliżenie czasu trwania okazuje się błędne (patrz „1. technika biznesowa: klient jest częścią zespołu" z rozdziału 2.).
- *Określenie kolejności wprowadzania funkcji* przez wybór kart funkcji do wprowadzenia w aktualnej iteracji (patrz „2. technika biznesowa: zabawa w planowanie" z rozdziału 2.).
- *Pomiar postępów prac nad projektem* w dowolnym momencie przez wykonanie testów akceptacyjnych (patrz „1. technika tworzenia: kreowanie z nakierowaniem na testy" z rozdziału 2.).
- *Zakończenie projektu w dowolnym momencie bez utraty zysków*. Wynika to z faktu, że oprogramowanie jest utrzymywane w stanie pełnej gotowości do wydania i zawiera najbardziej przydatne funkcje (patrz „4. technika tworzenia: ciągła integracja" i „2. technika biznesowa: zabawa w planowanie", oba w rozdziale 2.).

Odpowiedzialność klienta

XP określa kilka elementów, za które odpowiada klient.

- *Podejmowanie decyzji technicznych powinien pozostawić programistom*, ponieważ to oni znają się na technologii (patrz „2. technika biznesowa: zabawa w planowanie" z rozdziału 2.).
- *Poprawna analiza ryzyka*, czyli poprawne uszeregowanie funkcji pod kątem ich znaczenia (patrz „2. technika biznesowa: zabawa w planowanie" z rozdziału 2.).
- *Wybór funkcji o największej wartości*. Wstawianie do harmonogramu następnej iteracji najważniejszych funkcji (patrz „2. technika biznesowa: zabawa w planowanie" z rozdziału 2.).

- *Precyzyjne opisanie funkcji*, aby programiści mogli wykonać dokładne karty zadań i poprawnie określić przybliżony czas ich wprowadzenia (patrz „Karty funkcji" z rozdziału 4.).
- *Praca w zespole*. Służenie radą i szybkie odpowiadanie na pytania (patrz „1. technika biznesowa: klient jest częścią zespołu" z rozdziału 2.).

Programista

Większość technik XP dotyczy codziennej pracy nad kodem. Zadaniem programisty jest zamiana opisu funkcji dostarczonego przez klienta na działający kod.

Rola programisty w planowaniu i implementacji funkcji zależy od jego wiedzy i rozumienia problemów technicznych. Programista tworzy i zarządza ewoluującym systemem. Musi odpowiadać na pytania: „W jaki sposób to zaimplementować?", „Ile czasu to zajmie?" i „Jakie jest ryzyko?".

Programista współpracuje z klientem, by dobrze zrozumieć opis funkcji. Na podstawie opisu decyduje o implementacji. Następnie określa, ile czasu zajmie wprowadzenie funkcji, bazując na zaproponowanej implementacji i zdobytym doświadczeniu. Oszacowanie czasu trwania pomaga klientowi umieścić w harmonogramie najważniejsze funkcje i odpowiedzieć na pytanie *jak długo?*.

Gdy tworzy się karty zadań z kart funkcji lub implementuje funkcję, można odkryć wzajemne zależności między funkcjami, ryzykowane funkcje wykorzystujące nową technologię lub zwiększoną złożoność zagadnienia. Programista omawia problemy z klientem, który może zmienić harmonogram. Zazwyczaj ryzyko jest niewielkie — redukuje je praktykowanie prostoty.

Prawa programisty

XP określa kilka praw programisty.

- *Określenie czasu potrzebnego na wprowadzenie funkcji*, czyli umożliwienie podejmowania decyzji technicznych (patrz „2. technika biznesowa: zabawa w planowanie" z rozdziału 2.).
- *Praca zgodnie z sensownym i przewidywalnym harmonogramem*. Harmonogram powinien zawierać tylko taką liczbę zadań, jaką można wykonać w rozsądnym przedziale czasu (patrz „4. technika biznesowa: praca we względnym spokoju" z rozdziału 2.).
- *Wykonanie kodu spełniającego potrzeby klienta*, dzięki skupieniu się na testowaniu, refaktoryzacji i komunikacji z klientem (patrz „1. technika tworzenia: kreowanie z nakierowaniem na testy" i „2. technika kodowania: bezlitosna refaktoryzacja", oba z rozdziału 2.).
- *Unikanie podejmowania decyzji biznesowych*. Powinien podejmować je klient (patrz „1. technika biznesowa: klient jest częścią zespołu" z rozdziału 2.).

Odpowiedzialność programisty

XP określa zakres odpowiedzialności programisty.

- *Stosowanie standardów zespołu*, aby system był prosty, dobrze przetestowany i sprawny (patrz rozdział 6.).
- *Implementacja tylko wymaganych funkcji*, by projekt był prosty i najbardziej cenny dla klienta (patrz rozdział 6.).
- *Stała komunikacja z klientem* w celu zrozumienia zagadnień i wykonania odpowiedniego harmonogramu (patrz „1. technika biznesowa: klient jest częścią zespołu" z rozdziału 2.).

Dodatkowe role

XP identyfikuje dwie dodatkowe role. Nie muszą się one pojawić w każdym zespole.

Organizator

Organizator zajmuje się śledzeniem zgodności z harmonogramem. XP stosuje kilka miar. Najważniejszą z nich jest szybkość zespołu, czyli stosunek oszacowanego czasu idealnego do rzeczywistego czasu spędzonego nad implementacją. Innym ważnym wskaźnikiem może być zmiana szybkości, ilość pracy wykonywanej ponad normę i stosunek testów z poprawnymi wynikami do testów z błędami.

Wszystkie z tych miar dotyczą postępu prac. Pomagają określić, czy projekt jest wykonywany zgodnie z harmonogramem iteracji. Umożliwiają wykrycie sygnałów, wskazujących na niezgodność z harmonogramem. Przyjrzenie się samym liczbom nie zawsze da pełny obraz sytuacji. Anomalie należy przedyskutować na spotkaniu z całym zespołem (patrz podrozdział „Iteracja" z rozdziału 3.).

Pomiar szybkości w iteracji powinien być codzienne lub co dwa dni. Organizator pyta się programistów, ile zadań udało im się wykonać. Zadawanie tych pytań powinno być wykonywane osobiście, ale nieformalne i w komfortowej atmosferze. Szczerość to podstawa, a organizator nie powinien być stronniczy. Organizatorem powinien być menedżer lub zaufany programista. Regularne sprawdzanie postępów pomaga przezwyciężyć problemy i pracować bardziej płynnie.

Trener

Niektóre projekty XP zawierają trenera, który doradza i mentalnie wspiera zespół. Taka pomoc jest szczególnie przydatna w trakcie

wprowadzania technik XP. Trener powinien świecić przykładem i być osobą wielce szanowaną przez resztę zespołu.

Czasem trudno w pełni stosować techniki XP. Choć wiele z nich ma sens, potrzeba czasu na dostosowanie się do nowych warunków. Pojawiają się przeszkody, które wymagają wsparcia mistrza. Trener powinien służyć swym doświadczeniem.

Trener stara się poprawić techniki XP i ogólne zasady tworzenia oprogramowania u programistów. Czasem uczy bezpośrednio. Innym razem sam siada przed komputerem i uczy na przykładach. Może zasugerować konkretną implementację, podsunąć rozwiązanie trudnego problemu technicznego, a także służyć jako pośrednik między zespołem i zarządem.