

# Programowanie eXtremalne

Adrian Gadzina

# XP – czym jest?

**Programowanie ekstremalne** (ang. eXtreme Programming, XP) to paradygmat i metodyka programowania mająca na celu wydajne tworzenie małych i średnich "projektów wysokiego ryzyka", czyli takich, w których nie wiadomo do końca, co się tak naprawdę robi i jak to prawidłowo zrobić. Przyświeca temu koncepcja prowadzenia projektu informatycznego, wywodząca się z obserwacji innych projektów, które odniosły sukces. - *Wikipedia*



# XP – czym jest?

- Opiera się ona na zbiorze zasad i sugestii, które powinny być praktykowane.
- Metodyka ta zastosowana w małych i średnich zespołach może przynieść ogromne korzyści. Często programiści stosują programowanie ekstremalne nie zdając sobie nawet z tego sprawy.
- Pomimo pewnych reguł, których trzeba przestrzegać, XP nie nakłada na programistów dodatkowej pracy nie związanej bezpośrednio z tworzeniem i pielęgnowaniem kodu programu.

# XP – czym jest?

- Lekkość tej metodyki oznacza, że rezygnuje ona z formalizmów, które często nadmiernie obciążają programistów i kierowników zespołów. Tutaj nie zmusza się ludzi do tworzenia obszernych stron dokumentów, których nikt nigdy nie przeczyta. **Podstawą jest robienie tylko tego, co jest w danej chwili potrzebne.**
- XP jest zaprojektowane w taki sposób, by wszystkie zasady uzupełniały się wzajemnie. Dzięki temu pomimo braku ściśle ustalonych formuł doprowadza do celu nie tylko w zamierzonym czasie, ale i z produktem **najwyższej jakości.**

# XP – czym jest?

- Jednak lekkość ta jednocześnie niekoniecznie oznacza, że XP łatwo jest używać w praktyce. Jak wykazują nawet nieformalne analizy działań firm, mało która jest w stanie sprostać wszystkim wymaganiom stawianym przez tę metodykę. Jednak utrzymanie dyscypliny pracy procentuje w tym, że osiąga się zadowolenie klienta oraz dużą satysfakcję z wykonanego zadania.

# Podstawowe praktyki XP

- Kent Beck (*twórca programowania ekstremalnego*) podkreśla jako kluczowe 12 praktyk, których powinno się przestrzegać, by można było powiedzieć, że realizuje się metodykę **XP**.
- W rzeczywistości trudno jest sprostać wszystkim wymaganiom.
- Proponowane są więc, podobnie jak w innych metodykach pewne stopnie, kolejne etapy, na drodze do pełnego stosowania XP.
- Trzeba podkreślić jednak, że dopiero stosowanie **wszystkich praktyk** jest w stanie zagwarantować **sukces** i zminimalizować szanse porażki.
- Wybór jest ekstremalny: albo pełna rewolucja i wielki sukces albo balansowanie pomiędzy **zabezpieczeniami** i **przziemnością**.

# Planowanie

- Tworzenie oprogramowania w XP odbywa się przyrostowo przez wdrażanie kolejnych wydań produktu.
- Tworzenie oprogramowania w XP odbywa się przyrostowo przez wdrażanie kolejnych wydań produktu.
- Do szacowania używa się jednostek zwanych idealnymi osobo-tygodniami. Idealny osobo-tydzień to tydzień pracy wyłącznie nad programem, bez dodatkowych zajęć, ale wliczający czas testowania programu.

# Planowanie

- Podczas gry planistycznej klient określa, które historie są dla niego najważniejsze i które z nich powinny być zrealizowane w pierwszej kolejności. W rezultacie powstaje spis funkcji systemu, które będą do niego dodawane w ramach kolejnych wydań produktu. Podczas tworzenia oprogramowania odbywa się wiele iteracji, z których każda jest oddzielnie planowana, ukończona złączeniem i osiągnięciem działającej wersji systemu.



# Małe wydania

- Małe kroki to częste łączenie kodu napisanego przez programistów
- Osiąga się je przez podział zadania na małe historie użytkownika. Dzięki temu pojedynczy fragment kodu może być łatwo i szybko wykonany, przetestowany i złączony z resztą systemu.
- Małe wydania to częste akceptacje powstałego systemu przez klienta. Dzięki ciągłym testom i łączeniu zawsze istnieje sprawnie działająca wersja, a klient nie musi długo czekać na kolejną. Ciągłe istnieje też informacja zwrotna od klienta, który ocenia czy zespół realizuje to, o co mu chodziło.

# Wspólny język

- Każdy zespół programistyczny musi kontaktować się z klientem. Często dochodzi do sytuacji, w której po godzinach rozmów nagle odkryto, że klienci i projektanci mówią o zupełnie różnych rzeczach. Wynika to oczywiście z tego, że jedni i drudzy posługiwali się innymi pojęciami.
- Wspólny język jest szczególnie ważny dla klientów, którzy nie są zapoznani z technologią komputerową i którzy nie mogą operować specyficznymi pojęciami.

# Wspólny język

- Przykłady:
  - „oczyszczenie treści z niepotrzebnego kodu” zamiast „usunięcie nadmiarowych oraz niesemantycznych znaczników HTML”
  - „sprawdzenie czy informacje przesyłane w formularzu są podane poprawne” zamiast „walidacja danych przesyłanych metodą POST”

# Prosty projekt

- XP zakłada, że wymagania klienta, rynku i sytuacja w branży ciągle się zmieniają. Nie ma więc sensu planować rozwiązań, o których nie wiadomo, czy zostaną wykorzystane w przyszłości.
- Celem XP jest jak najszybsze i najprostsze osiągnięcie satysfakcji klienta przez dostarczenie oprogramowania, spełniającego postawione wymagania.
- Jeśli klient chce dodać nową funkcjonalność musi stworzyć nową historię. Jej koszt i pozycja na liście ważności zostaje ustalona w wyniku kolejnej gry planistycznej.

# Ciągłe testowanie

- Ciągłe testowanie to podstawowe działanie podczas pisania programu w metodzie **XP**.
- Programista jeszcze przed napisaniem danej procedury tworzy kod, który ma testować.
  - W ten sposób wcześniej musi **pomyśleć o wszystkich rzeczach**, które mogą pójść nie po jego myśli.
  - Dzięki temu podczas pisania właściwego kodu procedury zabezpieczają przed tymi możliwościami.
  - Pisanie procedury testowej nie powinno jednak trwać zbyt długo i nie powinna być ona zbyt rozbudowana.

# Przerabianie

- Przerabianie (eng. refactoring) jest konieczne zaraz po przetestowaniu działającej procedury. Przerabianie to „poprawianie projektu istniejącego kodu”.
- Przerabianie może być przeprowadzone w celu uzyskania wielu różnych efektów. Jako najbardziej oczywiste wymienia się poprawienie wydajności działania procedury, oraz uzyskanie lepszej struktury systemu.
- . Przykładowe konkretne działania podczas przerabiania to: skracanie metod, skracanie klas, usuwanie 'prawie' powtarzających się fragmentów kodu, usuwanie niepotrzebnych iteracji, usuwanie zbyt wielu zmiennych roboczych.
- Każdorazowe przerabianie, nawet jeśli jest najprostsze, musi oczywiście zakończyć się uruchomieniem testów.

# Programowanie w parach

- Programowanie w parach jest trudne, wymaga dobrego zgrania zespołu, ale przynosi wymierne korzyści w postaci lepszego kodu. Programowanie w parach pomaga również w dokonywaniu poprawek. Druga osoba może bowiem wiedzieć więcej o danym fragmencie kodu.
- Generalnie programowanie w parach pomaga propagować wiedzę o różnych fragmentach programu na całą grupę osób, systemie.



# Programowanie w parach

- Każdy kto kiedykolwiek spróbował programować w parach doświadczył, że diametralnie zmienia ono sposób pisania kodu. Podczas gdy jedna osoba (trzymająca klawiaturę) pisze kod, druga na bieżąco go sprawdza, sugeruje możliwe rozwiązania, może służyć pomocą i zwraca uwagę na błędy.
- Tak powstały kod jest nie tylko lepszy ale i łatwiej oraz szybciej się kompiluje. Według Kenta pary powinny się między sobą mieszać. Również programiści wewnątrz pary powinni co jakiś czas zamieniać się miejscami. w ten sposób wysiłek jest rozłożony równomiernie.





# Standard kodowania

- XP narzuca wszystkim programistom wspólny standard kodowania i dokumentowania. Standard taki powinien być ustalony i zaakceptowany przez całą grupę.
- Standard powinien jednoznacznie określać wygląd kodu, ale nie powinien być zbyt długi i szczegółowy.
- Standard dokumentowania zakłada, że samych komentarzy w kodzie jest jak najmniej. Klasy powinny być tak zaprojektowane by przeznaczenie poszczególnych metod było jasne, a samo działanie oczywiste.
- XP stara się podtrzymać naturalne skłonności programistów do upiększania i ulepszenia kodu

# Wspólna odpowiedzialność

- Kiedy trzeba szybko wykonać poprawki nie ma czasu na poszukiwania „właściwej osoby”. Taka osoba może być zresztą już nieosiągalna. W XP wszyscy są odpowiedzialni tak samo.
- Jeśli trzeba coś zmodyfikować nie ma problemu, bo poprawki może zrobić każdy. Częste przeorganizowywanie doprowadza kod do stanu dobrej przejrzystości, a gotowe procedury testujące zapewniają, że poprawki nie doprowadzą do katastrofy.
- XP preferuje umieszczenie całej grupy programistów w jednym pomieszczeniu, co ma pomagać w komunikacji i rozwijaniu życia grupy.

# Ciągłe łączenie

- Ciągłe łączenie to integracja programu tak często, jak to tylko możliwe.
- Programista po wykonaniu każdego nowego fragmentu programu łączy go z systemem.
- Najczęściej stosuje się jedną maszynę, na której w danej chwili może pracować jedna osoba zajmująca się łączeniem kodu.
- Ciągłe łączenie jest ułatwione w XP dzięki prostym projektom, ciągłym testom i wspólnej odpowiedzialności za kod.

# 40-godzinny tydzień pracy

- Swego rodzaju symbolem, znakiem rozpoznawczym XP, stało się wymaganie 40-to godzinnego tygodnia pracy.
- Zespoły programistów powinny być przyzwyczajone do **stałej wydajności i stałego obciążenia**.
- Może przytrafić się czasem jeden tydzień nieco większego obciążenia, ale dwa tygodnie mogą już oznaczać kłopoty z harmonogramem prac.
- Oczywiście jest, że dla niektórych zespołów tydzień może trwać 45 godzin, a dla innych 35. Istotne jest to, by ustalić **konkretną**, nienaruszalną granicę obciążenia grupy.

# Ciągły kontakt z klientem

- Aby zadowolić wymagania klienta należy bezwzględnie podążać za jego życzeniami.
- XP zakłada ciągłą możliwość konsultacji z klientem „na żywo”. W praktyce oznacza to codzienną obecność klienta w zespole programistów. Często bywa to jednak trudne do spełnienia. Zespoły takie mogą zrezygnować z XP, zorganizować zastępczą formę komunikacji z klientem, bądź znaleźć inne rozwiązania pozwalające na utrzymanie więzi z odbiorcą systemu.

# Podsumowanie

- Programowanie Ekstremalne jest przykładem lekkiej metodyki, przyjaznej zarówno klientowi, jak i programistom.
- Zestaw dwunastu specyficznych dla XP praktyk, choć istotnie różniących się od metod znanych z tradycyjnej inżynierii oprogramowania, ma za zadanie osiągnięcie tych samych celów. Są one ukierunkowane na różne obszary aktywności w procesie budowy oprogramowania, od budowania relacji z klientem i pozyskiwania i zarządzania wymaganiami, poprzez projektowanie i testowanie aż do tworzenia kodu, które jest jądrem XP.

# Podsumowanie

- Kanon praktyk podanych przez Kenta Becka na razie stanowi wzorzec wszelkich implementacji XP. Powstają modele próbujące klasyfikować lekkie procesy budowy oprogramowani i porównywać je z metodykami klasycznymi, jednak trudno na razie przesądzać o roli, jaką może odegrać XP.
- Niestety, XP posiada także kilka wad. Istotnym aspektem jest koszt wdrożenia takiej metodyki. Z przeprowadzonych badań wynika, że samo wdrożenie programowania parami jest znacznie kosztowniejsze niż zatrudnianie programistów pracujących pojedynczo.

# Podsumowanie

- Nieznana jest także skuteczność stosowania XP przy tworzeniu systemów o wysokiej niezawodności oraz wymagających ścisłego dotrzymania harmonogramu. Wdrożenie XP napotyka często opory ze strony decydentów, którzy odejście od planowanego i udokumentowanego procesu kojarzą ze spadkiem jakości produkowanego oprogramowania.
- Programowanie Ekstremalne wymaga dopasowania do okoliczności, w jakich jest stosowane, jednak faktycznie wnosi istotne nowości oraz niezbędny powiew świeżości. Oznacza to, że XP stanowi, lub niedługo będzie stanowić, istotną alternatywę dla tradycyjnych metod inżynierii oprogramowania, jednak obecnie wymaga jeszcze wielu badań i eksperymentów



# Koniec

Dziękuję za uwagę

