Feature Driven Development

lekka metodyka tworzenia oprogramowania

Kasprzyk Andrzej IS II

Wstęp

Feature Driven Development (FDD) to metodyka tworzenia oprogramowania, która wspomaga zarządzanie fazami analiz, projektowania i konstrukcji oprogramowania. Główną cechą FDD jest realizacja projektu w krótkich iteracjach wynikających z wymagań użytkownika. FDD została opracowana w 1998 roku przez Jeffa De Luca i Petera Coada. FDD jest, obok *eXtreme Programming, Scrum, Agile Modeling*, przedstawicielem tzw. lekkich metodyk tworzenia oprogramowania, które w ostatnich latach zdobywają coraz większą popularność. Metody te, kładące nacisk na komunikację między członkami zespołu i klientami oraz elastyczność w doborze technik, pozwalają na szybsze dostarczanie zamówionego produktu przy zachowaniu odpowiedniej jakości.

Istota FDD

Głównym elementem FDD jest pojęcie cechy (ang. *feature*) produktu. Cecha to mały fragment funkcjonalności produktu, mający wartość z punktu widzenia klienta, tzn. dostarczający interesujących dla niego wyników. Cecha powinna być na tyle „mała”, aby czas jej realizacji nie przekraczał 2 tygodni. Cechy są sposobem opisu wymagań funkcjonalnych klienta.

Kluczowe role w FDD

- menadżer projektu,
- eksperci dziedzinowi,
- główny architekt,
- menadżer programistów,
- główni programiści,
- właściciele klas.

Menadżer projektu

jest odpowiedzialny za całość projektu. Zarządza zakresem, harmonogramem oraz zasobami.

Eksperci dziedzinowi

są nimi użytkownicy, klienci, sponsorzy, analitycy biznesowi lub dowolne ich połączenie. Ich zadaniem jest przekazanie programistom wiedzy na temat wymaganej funkcjonalności systemu.

Główny architekt

jest odpowiedzialny za ogólny projekt systemu. Jest to techniczna funkcja wymagająca doskonałych umiejętności technicznych i analitycznych, jak również zdolności komunikacyjnych z ekspertami i pozostałymi członkami zespołu projektowego.

Menadżer programistów

jest odpowiedzialny za zarządzanie całym zespołem programistów. Jego głównym zadaniem jest rozwiązywanie konfliktów o zasoby między pracującymi współbieżnie podzespołami.

Główny programista

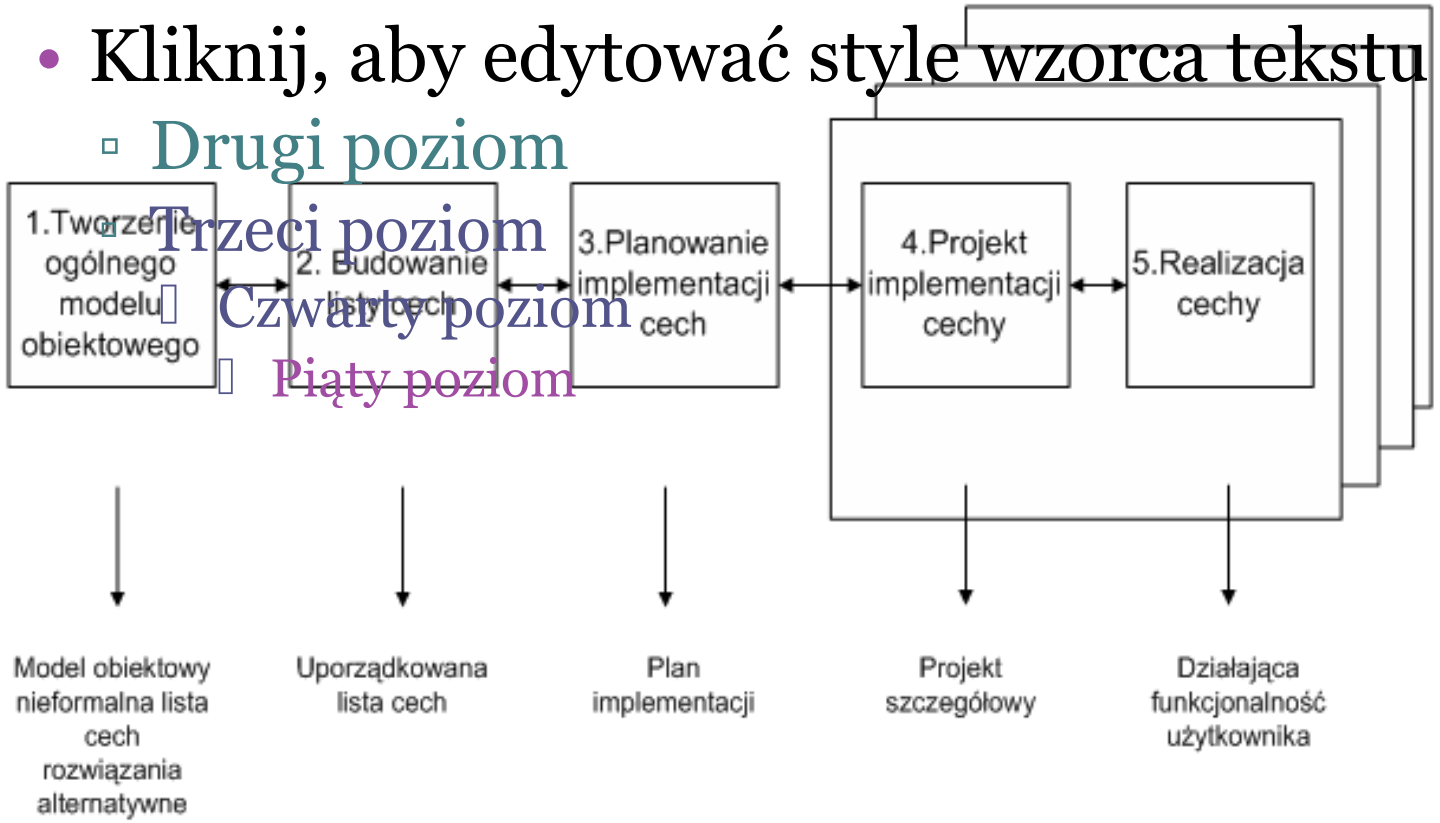
jest doświadczonym programistą, który jest odpowiedzialny za implementację przydzielonego mu zbioru cech produktu. Bierze aktywny udział w analizie wymagań oraz projektowaniu architektury rozwiązania. Kieruje podzespołem złożonym z 3-6 programistów przypisanych do realizacji danego zbioru cech.

Właściciele klas

to programiści, którzy pracują pod kierownictwem głównego programisty. Ich zadaniem jest projektowanie szczegółowe, kodowanie, testowanie i dokumentowanie cech produktu.

Fazy procesu FDD

- Kliknij, aby edytować style wzorca tekstu
  - Drugi poziom
    - Trzeci poziom
      - Czwarty poziom
        - Piąty poziom



Fazy procesu FDD

- tworzenie ogólnego modelu,
- budowanie listy cech,
- planowanie implementacji cech,
- projektowanie,
- implementacja.

Faza 1- Tworzenie ogólnego modelu

Zadaniem tego etapu jest stworzenie ogólnego modelu funkcji biznesowych realizowanego produktu (ang. *an overall model*). Ogólny model powinien dostarczać wiedzę o wszystkich wymaganych funkcjach bez specjalnego zagłębiania się w szczegóły. Jest on wspólnym dziełem analityków i programistów oraz przyszłych użytkowników, którzy są najlepszymi ekspertami z dziedziny produktu. Zaangażowanie użytkownika ma kluczowe znaczenie dla sukcesu projektu. Pozwala na wspólne rozumienie problemów przez użytkowników i programistów oraz minimalizuje „niejawne" założenia czynione prze jedną lub drugą stronę.

Tworzenie ogólnego modelu cd.

- stworzenie zespołu projektowego pod kierownictwem Głównego Architekta,
- przeprowadzenie przeglądu dziedziny problemu,
- studiowanie dokumentów z wymaganiami i z dziedziny problemu,
- przygotowanie alternatywnych modeli w oddzielnych małych grupach projektowych,
- wypracowanie wspólnego modelu,
- zatwierdzenie ogólnego modelu,
- zadokumentowanie istotnych założeń dotyczących projektu i alternatywnych rozwiązań.

Faza 2 – Budowanie listy cech

W oparciu o model opracowany w fazie 1 tworzona jest lista cech produktu (ang. *feature list*), które zapewnią dostarczenie wymaganej przez klienta funkcjonalności. W pierwszym kroku wyodrębnia się obszary przedmiotowe odpowiadające głównym fragmentom funkcjonalnym. Następnie każdy obszar przedmiotowy jest dzielony na poszczególne aktywności, czyli zbiory cech. Każdy krok w aktywności jest identyfikowany jako cecha. W wyniku tego kroku powstaje hierarchicznie uporządkowana lista cech produktu.

Faza 3 – Planowanie implementacji cech

Ma na celu przygotowanie planu określającego w jakiej kolejności cechy będą implementowane (ang. *plan by feature*). W tym procesie brane są pod uwagę takie czynniki jak priorytet danej cechy, zależności między cechami, złożoność implementacyjna oraz stopień obciążenia zespołu programistów. Ważnym kryterium jest również podział całego projektu na „zewnętrznie" obserwowalne etapy, czyli kroki milowe (ang. *milestones*) w projekcie. Efektem tej fazy jest plan implementacji, który określa datę (miesiąc, rok) realizacji każdego zbioru cech. Za każdy zbiór cech jest odpowiedzialny wyznaczony Główny Programista. Znany jest również przydział poszczególnych klas programistom, którzy będą je realizowali. Realizacja zadań w tej fazie składa się z:

Planowanie implementacji cech cd.

- sformowania zespołu planującego,
- określenia kolejności implementacji,
- przypisania zbioru cech do Głównych Programistów,
- przypisania klas do programistów.

Faza 4 i 5 – Projektowanie i implementacja

Istotą FDD jest dostarczanie kolejnych, „działających” wersji produktu w krótkich iteracjach składających się z projektowania szczegółowego (ang. *design by feature*) oraz implementacji (ang. *build by feature*) wybranego zbioru cech produktu. Cechy zakwalifikowane do realizacji w danej iteracji są przydzielane do realizacji dynamicznie tworzonym zespołom programistów (ang. *feature team*). Taki mały zespół typowo składa się z 2-3 programistów. Zadaniem zespołu jest implementacja małego zbioru cech. Kod danej cechy jest pisany przez członka zespołu, któremu została przypisana klasa biznesowa związana z funkcjonalnością danej cechy. Napisany kod podlega przeglądowi przez pozostałych członków zespołu. Po pomyślnym wykonaniu testów jednostkowych nowy kod nadaje się do integracji z resztą produktu, który staje się bogatszy o kolejną cechę.

Projektowanie

- sformowanie zespołu programistów pod kierunkiem Głównego Programisty,
- opcjonalny przegląd dziedziny problemu i studiowanie dokumentów referencyjnych,
- stworzenie diagramów sekwencji (ang. *sequence diagram*),
- uszczegółowienie modelu obiektowego,
- zapisanie nagłówków klas i metod,
- inspekcja projektu.

Implementacja

- implementacja kodu klas,
- przeprowadzenie inspekcji kodu,
- testowanie jednostkowe,
- integracja nowego kodu z produktem.

Najlepsze praktyki

FDD podobnie jak inne lekkie metody bazuje na zbiorze najlepszych praktyk (ang. *best practices*), których stosowanie w połączeniu ze zdefiniowanym procesem tworzenia oprogramowania zapewnia efektywne wykonanie projektu. Kluczowe dla FDD są następujące praktyki:

Najlepsze praktyki cd.

- oparcie procesu o wymagania klienta,
- architektura systemu,
- krótkie iteracje,
- indywidualna odpowiedzialność za kod,
- inspekcje,
- regularne budowanie produktu,
- zarządzanie konfiguracją,
- raportowanie i widoczność wyników.

Podsumowanie

W pracy przedstawiono Feature Driven Development. FDD bazuje na uznanym zbiorze najlepszych praktyk promowanych jako panaceum na problemy z efektywną realizacją projektów informatycznych. Dobrze zdefiniowany proces bardzo mocno zorientowany na implementację oczekiwanej przez klienta funkcjonalności, precyzyjnie opisane role i odpowiedzialności członków zespołu oraz wsparcie dla zarządzania projektem czyni FDD godnym polecenia dla firm chcących usprawnić swój proces tworzenia oprogramowania. Przydatność FDD potwierdziła się w wielu projektach realizowanych również przez autorów metodyki, które zakończyły się sukcesem.

Dziękuję za uwagę.