Waterfall model

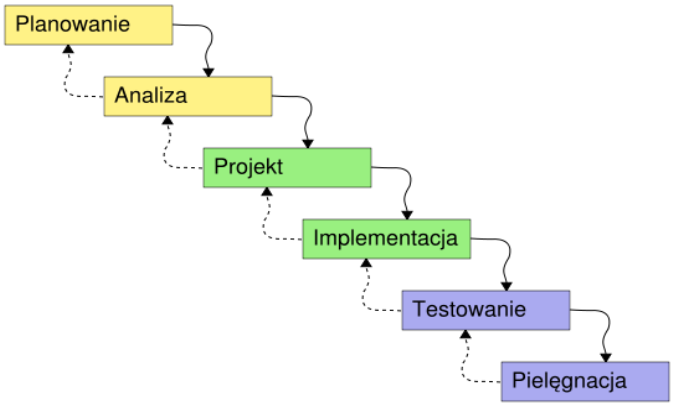
(iteracyjny model kaskadowy)

Marcin Wilk

Iteracyjny model kaskadowy – jeden z kilku rodzajów procesów tworzenia oprogramowania zdefiniowany w inżynierii oprogramowania. Jego nazwa wprowadzona została przez Winstona W. Royce w roku 1970, w artykule "Managing the Development of Large Software Systems" (zarządzanie tworzeniem dużych systemów informatycznych).

Polega on na wykonywaniu podstawowych czynności, jako odrębnych faz projektowych, w porządku jeden po drugim. Każda czynność to kolejny schodek (kaskada):

1. Planowanie systemu (w tym specyfikacja wymagań)
2. Analiza systemu (w tym Analiza wymagań i studium wykonalności)
3. Projekt systemu (poszczególnych struktur itp.)
4. Implementacja (wytworzenie kodu)
5. Testowanie (poszczególnych elementów systemu oraz elementów połączonych w całość)
6. Wdrożenie i pielęgnacja powstałego systemu.



Jeśli któraś z faz zwróci niesatysfakcjonujący produkt cofamy się wykonując kolejne iteracje aż do momentu, kiedy otrzymamy satysfakcjonujący produkt na końcu schodków.

Model kaskadowy jest rzadko używany z następujących powodów:

- Nie można przejść do następnej fazy przed zakończeniem poprzedniej
- Model ten posiada bardzo nieelastyczny podział na kolejne fazy
- Iteracje są bardzo kosztowne - powtarzamy wiele czynności

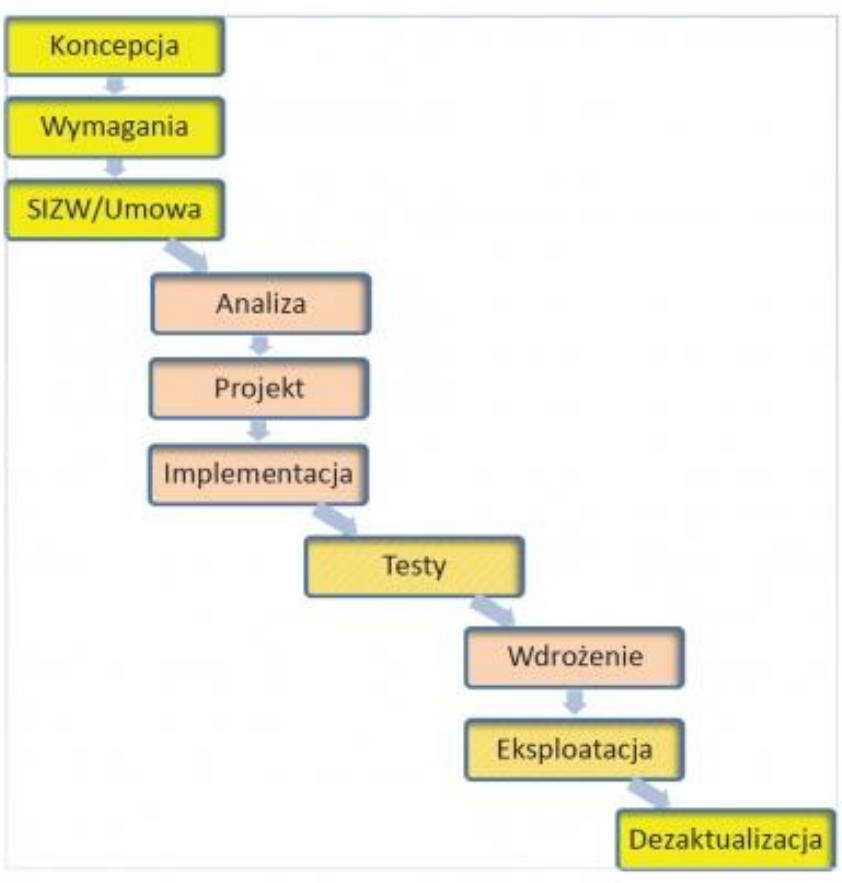
Tego typu modelu należy używać wyłącznie w przypadku, gdy wymagania są zrozumiałe i przejrzyste, ponieważ każda iteracja jest czasochłonna i wymaga dużych wydatków na ulepszanie.

Model kaskadowy w informatyce

Oprogramowanie jest produktem

Koncepcja modelowania cyklu życia oprogramowania wywodzi się z inżynierii oprogramowania. Przyjmuje się, że oprogramowanie jest szczególnym rodzajem wyrobu, którego wytwarzanie wymaga specjalnych procedur. Są one specyficzne dla każdego etapu cyklu życia. Pojęcie cyklu życia systemu informatycznego jest kluczowe dla zarządzania pracami wytwórczymi oprogramowania systemów informatycznych. W międzynarodowej normie IEC 60300 przedstawiono cykl życia oprogramowania według schematu, przedstawionego na rysunku. Nakładanie się na siebie niektórych elementów z tego rysunku oznacza, że kolejne fazy cyklu życia są częściowo realizowane w tym samym czasie, a także to, że pomiędzy tymi fazami może istnieć sprzężenie przyczynowo- skutkowe. Kolejność faz cyklu życia jest oznaczona umowną osią czasu i symboliką przesłaniania prostokątów, w których umieszczono nazwy poszczególnych faz.

Warto również przypomnieć, że oprogramowanie musi spełniać określone przez urząd wymogi jakościowe. W normie PN-ISO 8492 określono, że „jakość to ogół cech i właściwości wyrobu lub usługi decydujących o zdolności tego wyrobu lub usługi do zaspokojenia stwierdzonych lub przewidywanych potrzeb użytkownika produktu”. Oznacza to, że w interesie zarówno urzędu, jak i dostawcy oprogramowania, jest ustalenie wymogów jakościowych oraz doprowadzenie do ich należytego spełnienia.



Idea modelu kaskadowego

W podejściu kaskadowym kolejne etapy wytwarzania oprogramowania następują kolejno po sobie. Najpierw ustalamy potrzebę wytworzenia systemu informatycznego i wstępnie ustalamy, co to ma być. Potem dopracowujemy wymagania na system informatyczny i w ten sposób tworzymy opis przedmiotu zamówienia. Następnie opracowujemy pełną specyfikację istotnych warunków zamówienia (SIWZ) i na tej podstawie ogłaszamy przetarg na wykonanie systemu informatycznego. Po rozstrzygnięciu przetargu przystępujemy do współpracy z wykonawcą.

W pierwszym kroku wykonywana jest analiza systemu. Oznacza to, że uszczegóławiamy nasze wymagania z SIWZ po to, aby na tej podstawie wykonawca mógł opracować specjalistyczne modele analityczne systemu. Na etapie analizy nadal ustalamy, co ma być zrobione. Ważne jest, aby ustalić wszystkie szczegóły niezbędne dla dalszych etapów procesu wytwórczego.

Modele analityczne stanowią podstawę dla wejścia w projektowanie systemu. Na tym etapie wytwórca ustala, w jaki sposób wykonać działający system zgodny z naszymi wymaganiami. Potem wytwórca przystępuje do implementacji oprogramowania. Powstają wtedy działające programy. Są one testowane i integrowane w zamówione komponenty programowe. Następnie wytwórca prosi zamawiającego o przeprowadzenie sprawdzenia wytworzonego oprogramowania. Jest to wykonywane podczas testów akceptacyjnych, przygotowanych wcześniej przez wytwórcę w porozumieniu z zamawiającym.

Po pozytywnym przejściu testów akceptacyjnych system informatyczny może być wdrożony do eksploatacji. W czasie trwania eksploatacji systemu może zostać zidentyfikowana potrzeba wytworzenia kolejnej, ulepszonej jego wersji. Wtedy po przejściu etapu dezaktualizacji rozpoczynamy kolejną kaskadę.

W modelu kaskadowym mamy, zatem następujące po sobie kroki wytwarzania oprogramowania:

1. **Koncepcja**
    1.1. Pomysł na system
    1.2. Ogólny zarys wymagań
2. **Wymagania ogólne**
    2.1. Cechy użytkowe systemu
    2.2. Ograniczenia techniczne, organizacyjne i prawne
    2.3. Przewidywany czas i budżet projektu
    2.4. Wymogi jakościowe na system

3. **SIWZ/Umowa**

3.1. Wymagania ogólne stanowią podstawę dla opisu przedmiotu zamówienia

3.2. Pozostałe dokumenty zgodnie z prawem zamówień publicznych

3.3. Podpisanie umowy na wykonanie systemu

4. Analiza

4.1. Uszczegółowienie i zamrożenie wymagań

4.2. Modele analityczne systemu

5. Projekt

5.1. Modele projektowe systemu

5.2. Niezbędne wytyczne dla etapu implementacji oprogramowania

6. Implementacja

6.1. Wytworzenie kodów programowych

6.2. Zintegrowanie komponentów w jedną całość

6.3. Przygotowanie systemu do testowania

7. Testowanie

7.1. Testy wewnętrzne systemu u wytwórcy

7.2. Testy akceptacyjne z zamawiającym

8. Odbiór i wdrożenie

8.1. Uzyskanie pozytywnego wyniku testów akceptacyjnych

8.2. Formalny odbiór systemu

8.3. Wdrożenie do eksploatacji

9. Eksploatacja

9.1. Eksploatacja wstępna przy asyście wykonawcy

9.2. Eksploatacja samodzielna przez zamawiającego

9.3. Wnioski odnośnie rozwoju systemu

10. Dezaktualizacja (ewentualnie)

10.1. Albo: ustalenie potrzeby wytworzenia kolejnej wersji systemu

10.2. Albo: wycofanie obecnej wersji z użycia bez planowania produkcji wersji następnej

Relacje pomiędzy urzędem i dostawcą informatyki

Dla poprawnego wytworzenia i uruchomienia systemu informatycznego kluczowa jest współpraca urzędu zamawiającego system i wykonawcy tego systemu. Przedstawiony na rysunku schemat cyklu kaskadowego pokazuje, że na początku pomysłów na system prace są realizowane po stronie zamawiającego. Oznaczone to zostało kolorem żółtym. Istotne jest, aby urząd potrafił ustalić potrzebę realizacji systemu i opisać jego koncepcję. Następnie konieczne jest zdefiniowanie wymagań ogólnych na system i odpowiednie przeprowadzenie procedury zamówienia publicznego. Po przeprowadzeniu procesu zamówienia na wykonanie usługi wytworzenia i dostarczenia oprogramowania, następuje podpisanie odpowiedniej umowy pomiędzy urzędem i wykonawcą systemu.

Potem następują kolejne trzy kroki oznaczone kolorem pomarańczowym, które są realizowane przez wykonawcę. Są to: analiza, projekt i implementacja systemu. Następnie realizowane są testy systemu. Zanim wykonawca poinformuje o gotowości do przekazania systemu, wykonuje on swoje testy wewnętrzne. Potem zamawiający przeprowadza testy akceptacyjne systemu. Ten etap cyklu wytwórczego jest bardzo ważny, ponieważ wyłącznie na podstawie pozytywnego przeprowadzenia testów akceptacyjnych może nastąpić odbiór systemu i tym samym płatność końcowa wynagrodzenia za jego wykonanie.

Kolejnym krokiem wykonywanym przez wytwórcę systemu jest jego wdrożenie na instalacjach docelowych i uruchomienie procesu eksploatacji systemu. Na etapie eksploatacji następuje sukcesywne przejmowanie odpowiedzialności za ciągłość pracy systemu przez urząd. Po zakończeniu okresu rękojmi i gwarancji urząd eksploatuje system, najczęściej całkowicie samodzielnie, bez wsparcia wykonawcy.

Ewentualny etap dezaktualizacji oprogramowania pozostaje w zakresie odpowiedzialności urzędu, który sam decyduje, o tym, czy i kiedy używane przez niego oprogramowanie przestało być wystarczające. Wtedy można podjąć decyzję o wprowadzeniu kolejnej, ulepszonej wersji oprogramowania. To z kolei uruchamia nowy cykl życia kolejnej wersji oprogramowania i historia zatacza koło.

Piramida propagacji błędów

W podejściu kaskadowym stosunkowo późno widzimy działające oprogramowanie. Wynika to z tego, że ogólne wymagania na system są ustalane przez zamawiającego na etapie przygotowywania SIWZ. Potem, po skutecznym przeprowadzeniu procedury zamówieniowej, następuje podpisanie umowy. Jeszcze na etapie analizy wstępnej następuje ostateczne uściślenie wymagań na system, a potem zamawiający oczekuje na wytworzenie systemu i możliwość przeprowadzenia jego testów akceptacyjnych.

Ułożenie relacji pomiędzy zamawiającym i wytwórcą w takim klasycznym – kaskadowym – podejściu rodzi wiele zagrożeń. Przede wszystkim może spowodować, że zamawiający nie będzie zadowolony z produktu, jaki został mu dostarczony. Wynika to głównie z tego, że wytwórca nie musi do końca dobrze rozumieć oczekiwań zamawiającego, sformułowanych w opisie przedmiotu zamówienia i na wstępnym etapie analizy systemu. Jeżeli mamy do czynienia z taką sytuacją, to nie jest dobrze. Mówimy wtedy o tym, że w końcowym produkcie systemu zostały zmaterializowane błędy z etapu definiowania wymagań na system

i z etapu analizy. Błędy te, siłą rzeczy, przenoszą się na dalsze etapy cyklu życia oprogramowania, co prowadzi do wytworzenia produktu nie do końca zgodnego z oczekiwaniami urzędu.

Sytuacja opisana powyżej jest znana w inżynierii oprogramowania, jako piramida propagacji błędów.

W czasie wytwarzania systemu nie możemy zapominać o istnieniu zjawiska piramidy propagacji. Z praktyki inżynierii oprogramowania wynika, że usunięcie błędu analitycznego z etapu specyfikowania wymagań może być nawet do 200 razy droższe (!) przy podstawie piramidy niż u jej wierzchołka. Konsekwencje błędów mogą być, więc ogromne. Warto tutaj również pamiętać o tym, że nie wszystkie błędy zlokalizowane zbyt późno można usunąć z oprogramowania. Wtedy mamy jeszcze większy problem.

Jak unikać lub osłabiać skutki piramidy propagacji błędów? Inżynieria oprogramowania zna różne metody przydatne w tym miejscu. Będziemy o nich pisać na łamach kolejnych numerów naszego pisma.