

Student: Jakub Woźniak pd4135
Kurs: Budowa i integracja systemów informacyjnych (BYT) - studia podyplomowe SAB
Prowadzący: Włodzimierz Dąbrowski

Wybrane modele cyklu życiowego wytwarzania oprogramowania dla firmy “Krasnale”

SPIS TREŚCI	Strona
1. Wstęp.....	2
1.1 Cel dokumentu.....	2
1.2 Zakres projektu.....	2
2. Przegląd wybranych modeli	3
2.1 Model kaskadowy.....	3
2.1.1 Charakterystyka.....	3
2.1.2 Realizacja w firmie “Krasnale”.....	4
2.1.3 Ocena.....	4
2.2 Model spiralny.....	4
2.2.1 Charakterystyka.....	4
2.2.2 Realizacja w firmie “Krasnale”.....	5
2.2.3 Ocena.....	5
2.3 Model UP.....	6
2.3.1 Charakterystyka.....	6
2.3.2 Realizacja w firmie “Krasnale”.....	6
2.3.3 Ocena.....	7
3. Wnioski końcowe.....	7

1. Wstęp

1.1 Cel dokumentu

Celem poniższej analizy jest porównanie trzech wybranych modeli cyklu życiowego wytwarzania oprogramowania (kaskadowego, spiralnego oraz Unified Process). Dokument zawiera krótki opis każdego modelu i propozycję realizacji dla konkretnego projektu informatycznego (firma "Krasnale"). Na koniec analizy spróbujemy porównawczo ocenić każde z tych podejść i wybrać optymalną metodę dla studium przypadku "Krasnale".

1.2 Zakres projektu

Według opisu przypadku pt. "Krasnale", firma ta zajmuje się dystrybucją krasnali ogrodowych renomowanych światowych producentów. Głównym celem projektu informatycznego byłoby dostarczenie narzędzi do sprzedaży przez Internet oraz unowocześnienie istniejącej strony WWW. Sponsorem projektu został wybrany dyrektor działu sprzedaży, który jest największym działem firmy "Krasnale". Zarząd oczekuje, że projekt będzie gotowy w ciągu 4 miesięcy.

Poniższa tabela zawiera ogólny zakres prac wraz z powstającymi artefaktami projektowymi i propozycją ich odbioru. Realizacja tych zadań oraz harmonogram będą się naturalnie różnić zależnie od tego, jaki przyjął się model wytworzenia oprogramowania. Szczegółowe propozycje realizacji zawarto w punktach 2.1.2 (model kaskadowy), 2.2.2 (model spiralny) oraz 2.3.2 (model UP).

Zadania	Artefakty projektowe	Metody odbioru
Prace analityczne		
Określenie wymagań dla: - sklepu internetowego (system składania zamówień z możliwością dalszej rozbudowy), - strony www, - systemu informacyjnego (promocje, ankiety, fora, listy mailingowe)	Raport wymagań funkcjonalnych i нефункциональных	Uzgodnione z klientem, zatwierdzone przez Sponsora i Zarząd.
Oszacowanie budżetu projektu	Budżet projektu	Uzgodnione z klientem, zatwierdzone przez Sponsora i Zarząd.
Oszacowanie harmonogramu projektu	Harmonogram prac z kamieniami milowymi	Uzgodnione z klientem, zatwierdzone przez Sponsora i Zarząd.
Analiza ram prawnych projektu	Raport ze skutków prawnych wdrożenia projektu	Zatwierdzone przez Sponsora i Zarząd.
Prace projektowe		
Projekt strony, sklepu internetowego i systemu informacyjnego	Szczegółowy model logiczny systemu zawierający m.in. diagramy przypadków użycia, klas, stanów i aktywności	Uzgodnione z klientem, zatwierdzone przez Sponsora i Zarząd.
Projekt architektury systemu	Projekt komponentów, implementacji, bazy danych, fizycznej struktury systemu	Zatwierdzone przez Sponsora.
Projekt graficzny	Projekt interfejsu użytkownika i szablonów HTML spójne z identyfikacją wizualną klienta	Uzgodnione z klientem, zatwierdzone przez Sponsora i Zarząd.
Implementacja		

Zaprogramowanie działającego sklepu internetowego, strony WWW i systemu informacyjnego	Działające oprogramowanie (front-end, back-end, gotowa baza danych)	Zatwierdzone przez Sponsora.
Testowanie		
Testy funkcjonalne i jednostkowe	Raport z testów (spełnienie wymagań, wydajność, niezawodność)	Zatwierdzone przez Sponsora.
Wdrożenie		
Szkolenie personelu, instalacja i przekazanie systemu klientowi	Dokumentacja szkoleniowa	Zatwierdzone przez Sponsora.

2. Przegląd wybranych modeli

2.1 Model kaskadowy

2.1.1 Charakterystyka

Model kaskadowy lub inaczej wodospadowy jest najstarszym i najprostszym sposobem na wytwarzanie oprogramowania. W tym podejściu budowa oprogramowania jest podzielona na poszczególne fazy, z których każda następuje liniowo dopiero po zakończeniu poprzedniej. W fazie wymagań określamy szczegółowe wymagania systemu. W fazie projektowania następuje szczegółowe projektowanie systemu. W fazie implementacji następuje kodowanie poszczególnych modułów w konkretnym środowisku i ich jednostkowe testowanie. W fazie testowania następuje testowanie całego oprogramowania. W fazie konserwacji następują czynności typowe dla utrzymania systemu informatycznego takie jak naprawa błędów i dodawanie nowych funkcji.

Ponadto, na wymienione powyżej fazy nakładają się jeszcze niezależnie:

- faza strategiczna, w której podejmujemy strategiczne decyzje przed rozpoczęciem realizacji projektu,
- faza analizy, podczas której budujemy model logiczny systemu,
- faza dokumentacji, w której równolegle z produkcją kodu wytwarzamy dokumentację projektową,
- faza wdrożeniowa, podczas której przekazujemy system użytkownikom końcowym.

Istnieje wiele odmian modelu kaskadowego, których celem jest minimalizowanie ryzyka wykrycia kosztownych błędów na późnym etapie. Jedną z nich jest tzw. model kaskadowy ze sprzężeniem. W tym podejściu wytwarzamy system kaskadowo z tym, że przed rozpoczęciem kolejnej fazy dajemy sobie szansę na korektę błędów m.in. poprzez częstszy kontakt z klientem i ew. powrót do poprzedniej fazy.

Główną zaletą tego podejścia jest łatwość planowania i zarządzania całym procesem. Wadami tego modelu jest duże ryzyko kosztownych błędów, które mogą być wykryte późno np. w fazie testowania lub konserwacji. Zwykle po ustaleniu wymagań kontakt z klientem staje się rzadszy lub w ogóle ustaje, co może zmniejszyć jego zainteresowanie końcowym produktem. Może się też okazać, że końcowe funkcjonalności różnią się z początkowymi oczekiwaniami klienta.

2.1.2 Realizacja w firmie "Krasnale"

Poniżej przedstawiono propozycję realizacji modelu kaskadowego w formie diagramu Gantta. Założyliśmy czasy realizacji poszczególnych zadań, biorąc pod uwagę skład zespołu, umiejętności jego członków i brak doświadczenia w realizacji podobnych zadań.

Zadania	Dni	12'23	1	2	3	4	01'24	1	2	3	4	02'24	1	2	3	4	03'24	1	2	3	4	04'24	1
Prace analityczne	33																						
wymagania dla sklepu internetowego	7																						
wymagania dla strony www	7																						
wymagania dla systemu informacyjnego	4																						
oszacowanie budżetu projektu	5																						
oszacowanie harmonogramu projektu	7																						
analiza ram prawnych projektu	3																						
Prace projektowe	25																						
projekt strony, sklepu i systemu informacyjnego	14																						
projekt architektury systemu	7																						
projekt graficzny	4																						
Implementacja	36																						
zaprogramowanie sklepu	14																						
zaprogramowanie strony	8																						
zaprogramowanie systemu informacyjnego	14																						
Testowanie	10																						
testy funkcjonalne i jednostkowe	10																						
Wdrożenie	10																						
szkolenie, instalacja i przekazanie systemu	10																						
ŁĄCZNIE:	114																						

2.1.3 Ocena

Oceniając symulację realizacji projektu dla firmy "Krasnale" w realizacji kaskadowej możemy stwierdzić, że dużą zaletą tego podejścia jest przejrzystość całego procesu. Każdy etap następuje kaskadowo dopiero po zakończeniu poprzedniego. Pomimo tego, nawet biorąc pod uwagę ostrożne planowanie czasu na poszczególne fazy możemy stwierdzić, że realizacja projektu w planowanym czasie 4 miesiące (posiedzenie rady nadzorczej) może być zagrożona. Jest to o tyle istotne, że w modelu kaskadowym testowanie końcowych produktów jest odsunięte w czasie, co podnosi ryzyko całego projektu. Możemy sobie wyobrazić sytuację, w której podczas testowania sklepu internetowego okaże się, że np. składanie zamówień przebiega inaczej niż w obecnym systemie lub generowane są dokumenty niezgodne ze standardami działu finansowego. Dodatkowe ryzyko zwiększa małe doświadczenie zespołu deweloperskiego w podobnych projektach.

2.2 Model spiralny

2.2.1 Charakterystyka

W modelu spiralnym oprogramowanie wytwarzane jest w sposób iteracyjno-kaskadowy. Wyróżniamy w nim cztery fazy, następujące po sobie we względnie krótkich cyklach. W fazie planowania, ustalamy ogólne cele budowy systemu (nowego lub kolejnej wersji istniejącego). W kolejnej fazie analizujemy ryzyko budowy systemu i ew. budujemy w tym celu prototyp. W fazie konstrukcji budujemy (implementujemy) właściwe oprogramowanie według podejścia kaskadowego. W fazie atestowania oprogramowanie jest oceniane przez klienta. Jeśli nie zostanie zaakceptowane, rozpoczyna się kolejny cykl. W efekcie, na koniec każdego cyklu zostaje wytworzony konkretny element oprogramowania.

Jest to typowy model wytwarzania oprogramowania rynkowego przez duże firmy np. Microsoft, Adobe, Oracle. Atestowanie jest dokonywane przez klientów używających te programy a nowe funkcje lub naprawa błędów ma miejsce w kolejnych wersjach oprogramowania.

Zaletą tego podejścia jest mniejsze ryzyko późnego wykrycia kosztownych błędów. Zorganizowanie takiego podejścia jest jednak w porównaniu z modelem kaskadowym bardziej

kosztowne m.in. poprzez większe przyzwoleń na eksperymentowanie (prototypowanie) i dużą liczbę iteracji. Podejście spiralne w praktyce dobrze sprawdza się w wytwarzaniu oprogramowania biznesowego. W takim przypadku mamy zwykle do czynienia z niejasnymi i / lub często zmieniającymi się wymaganiami i częstymi kontaktami z klientem. Wadę tego podejścia jest mniej rozbudowana dokumentacja.

2.2.2 Realizacja w firmie “Krasnale”

Poniżej przedstawiono propozycję realizacji modelu spiralnego w formie diagramu Gantta. Założyliśmy czasy realizacji poszczególnych zadań, biorąc pod uwagę skład zespołu, umiejętności jego członków i brak doświadczenia w realizacji podobnych zadań.

Zadania	Dni	12'23	1	2	3	4	01'24	1	2	3	4	02'24	1	2	3	4	03'24	1	2	3	4	04'24	1
Prace analityczne	18																						
analiza ram prawnych	3																						
wstępne wymagania dla strony i sklepu	5																						
analiza architektury dla strony i sklepu	5																						
wstępny projekt graficzny	5																						
Prototyp sklepu	20																						
doprecyzowanie wymagań	4																						
finalny projekt graficzny	4																						
programowanie	7																						
testy	5																						
Sklep internetowy	14																						
programowanie c.d.	7																						
testy c.d.	7																						
Wdrożenie sklepu	9																						
szkolenie użytkowników	5																						
przekazanie sklepu	4																						
Prototyp strony WWW	12																						
doprecyzowanie wymagań	2																						
finalny projekt graficzny	3																						
programowanie	5																						
testy	2																						
Strona WWW	7																						
programowanie c.d.	3																						
testy c.d.	4																						
System informacyjny	23																						
doprecyzowanie wymagań	3																						
finalny projekt graficzny	5																						
programowanie	6																						
testy	5																						
szkolenie użytkowników	4																						
ŁĄCZNIE:	96																						

2.2.3 Ocena

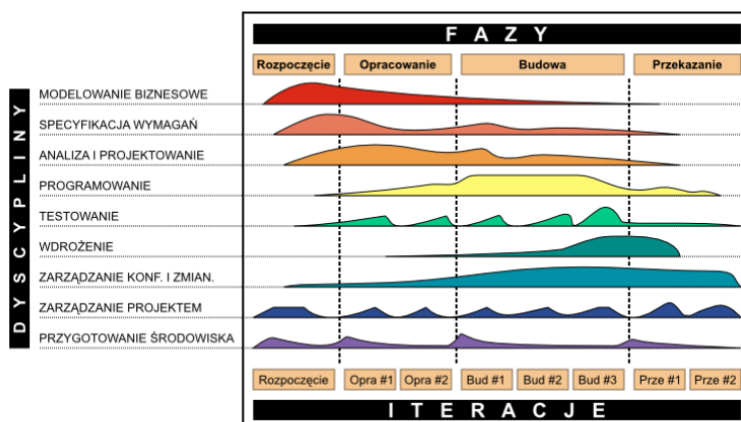
W porównaniu z modelem kaskadowym, realizacja w podejściu spiralnym charakteryzuje się większą liczbą zadań. Zwróćmy uwagę, że zadania podzielono na mniejsze podzadania, z których każde realizujemy w pętli (analiza-projektowanie/prototyp-implementacja-testy). Zwróćmy uwagę, że po zakończeniu każdego cyklu otrzymujemy widoczny efekt programistyczny (prototyp sklepu, prototyp strony), co zmniejsza ryzyko niezgodności oprogramowania z wymaganiami lub niedotrzymania terminów. Zakładając podobne nakłady na pracochłonność i ten sam zespół deweloperski, ostrożnie oceniamy, że w podejściu spiralnym mamy szansę na dotrzymanie 4-miesięcznego terminu.

2.3 Model UP

2.3.1 Charakterystyka

Rational Unified Process (RUP) to iteracyjny model wytwarzania oprogramowania opracowany przez firmę Rational Software Corporation (przejęte później przez IBM). Jest rozwinięciem modelu spiralnego i opiera się na metodyce iteracyjno-przyrostowej. Jest wykorzystywany w projektach informatycznych takich firm jak Visa, Lockheed-Martin, Volvo czy Intel.

Zakłada budowę systemu informatycznego w kolejnych iteracjach. Po zakończeniu każdego cyklu produkowany jest fragment systemu, który następnie jest udostępniany klientowi. Pozwala to znacznie zmniejszyć ryzyko całego projektu.



RUP wyróżnia cztery fazy cyklu życia projektu (rozpoczęcia, opracowywania, konstrukcji, przekazania). Zadania wykonywane podczas każdego cyklu należą do jednej z dziewięciu „dyscyplin” (modelowanie, wymagania, analiza, implementacja, testowanie, wdrożenie). Przejście do kolejnej fazy jest kamieniem milowym projektu, jednak brak spełnienia wymagań może spowodować konieczność powrotu do tej fazy i uzupełniania jej.

Zatem w podejściu RUP – odmiennie niż w modelu spiralnym - możliwe jest dokonywanie wielokrotnych iteracji wewnątrz każdej fazy tak, aby spełnić wymagania. Drugą różnicą w stosunku do modelu spiralnego jest większy nacisk na dokumentację projektową i ustrukturyzowanie całego procesu.

2.3.2 Realizacja w firmie “Krasnale”

Poniżej przedstawiono propozycję realizacji modelu Unified Process w formie diagramu Gantta. Założyliśmy czasy realizacji poszczególnych zadań, biorąc pod uwagę skład zespołu, umiejętności jego członków i brak doświadczenia w realizacji podobnych zadań.

Zadania	Dni	12'23	1	2	3	4	01'24	1	2	3	4	02'24	1	2	3	4	03'24	1	2	3	4	04'24	1
Rozpoczęcie	33																						
wymagania dla sklepu internetowego	7																						
wymagania dla strony www	7																						
wymagania dla systemu informacyjnego	4																						
oszacowanie budżetu projektu	5																						
oszacowanie harmonogramu projektu	7																						
analiza ram prawnych projektu	3																						
Opracowanie	25																						
projekt strony, sklepu i systemu informacyjnego	14																						
projekt architektury systemu	7																						
projekt graficzny	4																						
Budowa	36																						
zaprogramowanie sklepu	14																						
zaprogramowanie strony	8																						
zaprogramowanie systemu informacyjnego	14																						
Przekazanie	10																						
testy funkcjonalne i jednostkowe	10																						
szkolenie, instalacja i przekazanie systemu	10																						
ŁĄCZNIE:	104																						

2.3.3 Ocena

Symulacja realizacji projektu w oparciu o metodykę UP pozwala nam oczekiwać podobnego terminu realizacji co w modelu spiralnym. W odróżnieniu jednak od tego podejścia, metodyka UP jest bardziej skomplikowana i ustrukturyzowana, a większy nacisk kładzie się na jakość

końcowego produktu. Biorąc pod uwagę małe doświadczenie zespołu informatycznego w tego typu projektach, nie rekomendowałbym jednak jego realizacji w firmie "Krasnale".

3. Wnioski końcowe

Do wyboru optymalnego modelu cyklu życiowego zdecydowaliśmy się wesprzeć analizą ilościową każdej realizacji w postaci ważonej sumy punktów. Zdecydowaliśmy, że kryteriami wyboru, które będziemy oceniać według skali punktowej będą: szacunkowy czas realizacji, oczekiwana jakość oprogramowania, zgodność z wymaganiami biznesowymi, szacunkowy koszt, możliwość dalszej rozbudowy i łatwość wdrożenia. W treści studium przypadku podkreślone było, że oczekiwany czas realizacji i zgodność z wymaganiami będą najważniejszymi kryteriami wyboru, więc przyznaliśmy im największe wagi punktowe. Niższe wagi przyznaliśmy oczekiwanej jakości oprogramowania i oczekiwanemu budżetowi (jako istotne, ale nie decydujące), a najniższe możliwościom dalszej rozbudowy i łatwości wdrożenia. Tabelka przedstawia opis kryteriów, wagi oraz przyznane przez nas punkty dla każdego rozwiązania:

Kryterium oceny	Miara	Waga	Max punktów	Przyznane punkty wg realizacji:		
				kaskadowa	spiralna	UP
Przewidywany czas realizacji	[miesiące]	4	10	5	8	6
Oczekiwana jakość oprogramowania	[błędy / tydzień]	2	10	5	7	8
Zgodność z wymaganiami	[%]	3	10	5	7	8
Oczekiwany koszt	[PLN'000]	2	10	6	8	7
Możliwość dalszej rozbudowy	[%]	1	10	5	7	8
Oczekiwana łatwość wdrożenia	[%]	1	10	5	7	8
ŁĄCZNIE:			130	67	97	94

Biorąc pod uwagę tylko ocenę ilościową, powinniśmy odrzucić model kaskadowy, a zdecydować się na model spiralny lub UP. Skoro czas i spełnienie wymagań są kluczowe w tym projekcie, widzimy zbyt dużo ryzyk dla tego projektu w realizacji kaskadowej, co odzwierciedlają przyznane punkty.

Biorąc pod uwagę także wnioski z analizy jakościowej każdego projektu (par. 2.1.3, 2.2.3, 2.3.3) uważam, że powinniśmy zdecydować się na realizację w modelu spiralnym. Kluczową rolę w tym projekcie odgrywa czas (4 miesiące do posiedzenia rady nadzorczej). W realizacji w modelu spiralnym zespół programistyczny byłby w stanie pokazać swoim decydentom pierwsze prototypy najszybciej ze wszystkich podejść. Pozwoliłoby to szybko zebrać informację zwrotną o błędach i ewentualnej zmianie wymagań i dostosować projekt w kolejnej iteracji. Zwróćmy uwagę, że z treści studium przypadku możemy zakładać, że zespół programistyczny będzie prowadził projekt poza swoim codziennymi obowiązkami, takimi jak utrzymanie systemu finansowo-księgowego. Podnosi to dodatkowo ryzyko błędów i opóźnień w projekcie wynikających np. z potrzeby zgrania dokumentów księgowych pomiędzy sklepem internetowym a istniejącym systemem księgowym.

Jeśli jednak czas nie odgrywałby tak dużej roli w tym projekcie, realizacja według modelu UP mogłaby przynieść lepsze efekty w postaci lepszego dopasowania do wymagań, lepszej jakości i skalowalności oprogramowania.