

Zasady zarządzania projektem eLearningowym - Produkcja

Treść modulu

Moduł Zasady zarządzania projektem eLearningowym skupia uwagę na technikach prowadzenia projektów. W szczególności opisuje kluczowe fazy rozwoju projektu. Opisuje sposoby jego prowadzenia, podkreśla także znaczenie odpowiedniej dokumentacji potrzebnej do prowadzenia projektu.

Określenie celu edukacyjnego modulu

Celem modulu jest zapoznanie się bliżej z fazą produkcji projektu eLearningowego.

Podczas fazy produkcji będziemy precyzować mechanizmy zarządzania poszczególnymi działami projektowymi. Określimy dokumenty potrzebne do prawidłowego prowadzenia projektu, oraz te, które należy przygotować w celu przekazania produktu do klienta. Zwrócimy także uwagę na kwestie budżetu projektowego, wytłumaczymy jak zarządzać czasem pracowników i projektu. Na koniec opowiemy o sposobie przeprowadzania testów.



Wiedza bazowa potrzebna do pracy nad modulem

Informacje glównie

Produkcja - wstęp

Implementacja projektu eLearningowego, którą nazywamy fazą produkcji, składa się przede wszystkim ze stworzenia oprogramowania, a następnie zawarcia w nim klipów wideo, ścieżki audio, głosu lektora oraz animacji. W fazie produkcji tworzymy także dokumentację projektową. Materiały szkoleniowe, wdrożeniowe, manuale. Podręczniki użytkownika, administratora oraz wykładowcy. Poza tworzeniem oprogramowania, a fazy produkcji musimy zadbać o odpowiednie przetestowanie produktu, kontakty z klientem, akceptację kolejnych wersji (alfa/beta) oprogramowania. Zakończeniem fazy produkcji powinno być oddanie produktu klientowi.

Zarządzanie produkcją

Projekt multimedialny, niezależnie czy finalnie dostarczony na CD-ROM czy też poprzez Web, wymaga zespołu składającego się z wielu indywidualności, którym przyświeca wspólny cel. Podstawą dobrego zarządzania jest prawidłowa komunikacja pomiędzy członkami zespołu. Z reguły zarządzaniem zajmuje się kierownik projektu, który jest odpowiedzialny za "sklejenie" fragmentów projektu - w całość. Kierownik projektu musi zadbać także o zachowanie limitu budżetowego, oraz zakończenie kolejnych faz projektu zgodnie z ustalonym harmonogramem. Przy wdrażaniu dużych projektów, rola kierownika projektu może być ograniczona jedynie do badania wskaźników. (czas, koszty, komunikacja). Kierownik projektu nie zagłębia się wtedy w aspekty techniczne.

Zarządzanie czasem

Definicja "projektu" stanowi o "zawiazaniu się współpracy pomiędzy grupą ludzi, w celu osiągnięcia wspólnego celu, w określonym czasie". Niezależnie, czy w ustalonym czasie to przedsięwzięcie się powiedzie, czy też nie, projekt zakończy się z chwilą przekroczenia daty jego zakończenia.

Po tym wprowadzeniu powinniśmy zrozumieć, jak ważne jest więc prawidłowe zarządzanie czasem projektowym. Ustalenie terminu zakończenia projektu może być zgubne. Podczas pracy nad projektem, możemy renegeować czas jego zakończenia, jest to jednak bardzo trudne (szczególnie przy prowadzeniu projektów rządowych, zamówieniach publicznych, lub na przykład projektach, od których zależy ludzki los albo życie).

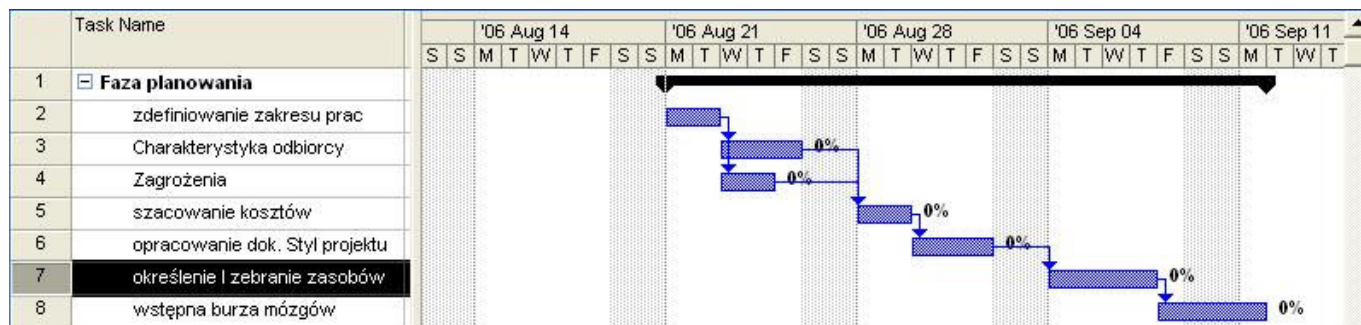
W przypadku projektów eLearningowych, nie będziemy mieli doczynienia z tak dużym zagrożeniem. Musimy sobie jednak zdawać sprawę, że umowa z klientem często zawiera punkt dotyczący kar umownych. Kary te, zobowiązany jest płacić kontrahent, który na czas nie wywiąże się z przekazania zamówionej usługi.

Najlepszą dla nas sytuacją jest taka, w której możemy spokojnie zaplanować czas potrzebny na wykonanie projektu, a następnie na podstawie tego planu podać czas zakończenia prac.

W przypadku pracy z klientem zewnętrznym, taka sytuacja jest bardzo żądka. Z reguły klient ma ustalony termin, przed którym chce otrzymać zamówione oprogramowanie. W takim przypadku kierownictwo projektu powinno zbadać, czy wykonanie prac w określonym przez klienta czasie jest możliwa. Czasami, by zachować termin część pracy przekazuje się do firm zewnętrznych, zatrudnia się pracowników sezonowych, albo w budżecie projektowym z góry (!!) zabezpiecza się środki finansowe na kary umowne.

W celu lepszego zobrazowania czasu projektowego używane są wykresy Gantta. Z ich pomocą możemy śledzić zależności pomiędzy różnymi czynnościami projektowymi.

	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	<input type="checkbox"/> Faza planowania	16 days	Mon 06-08-21	Mon 06-09-11		
2	zdefiniowanie zakresu prac	2 days	Mon 06-08-21	Tue 06-08-22		kierownik projektu
3	Charakterystyka odbiorcy	3 days	Wed 06-08-23	Fri 06-08-25	2	klient
4	Zagrożenia	2 days	Wed 06-08-23	Thu 06-08-24	2	kierownik projektu;klient
5	szacowanie kosztów	2 days	Mon 06-08-28	Tue 06-08-29	3;4	księgowy;kierownik projektu
6	opracowanie dok. Styl projektu	3 days	Wed 06-08-30	Fri 06-09-01	5	grafik;projektant
7	określenie i zebranie zasobów	4 days	Mon 06-09-04	Thu 06-09-07	6	metodyk
8	wstępna burza mózgów	2 days	Fri 06-09-08	Mon 06-09-11	7	kierownik projektu



Zarządzanie budżetem

Budżet, podobnie jak zarządzanie czasem, to podstawowa wartość którą podczas pracy nad projektem koniecznie trzeba monitorować!

Bardzo wiele projektów nie zostało zakończonych z powodu przekroczenia budżetu.

W przypadku kiedy mamy doczynienia z budową projektu na własne potrzeby, możemy zbudować budżet czasowy. Określimy na kiedy nasze oprogramowanie ma być gotowe, ile czasu na nie poświęcimy, a następnie postaramy się trzymać tych wstępnych wytycznych.

W większości projektów budżet budowany jest w fazie planowania lub projektowania aplikacji. W fazie planowania budżet pozwalał nam odpowiedzieć na pytanie czy projekt jest dla nas możliwy do wykonania, i czy nam się opłaca. W przypadku dużych firm, i dużych projektów z pewnością sprzedają najmie się negocjator, albo dział sprzedaży naszej firmy. My będziemy zobligowane do podania wstępnych wyliczeń czasowych, oraz o informacje dotyczące potrzebnych zasobów, środków, oprogramowania. Dział sprzedaży działa oczywiście na zasadzie generowania zysku. Tak więc, w przypadku projektu wartego 20 tys złotych, z którego negocjatorzy firmy chcieli by zarobić 40% kwoty, do dyspozycji pozostaje nam 12 tysięcy złotych. Dział sprzedaży należy w takim przypadku traktować jak osobną firmę, ich zadaniem było bowiem doprowadzić do podpisania kontraktu, i zebranie swojej marży. Nam pozostaje "zmieścić" się w pozostałej kwocie.

Zwrócimy też uwagę, w jaki sposób obliczać koszty dostępnych zasobów. Jeżeli wynagrodzenie grafika, który przygotowuje projekt oraz obiekty graficzne, wynosi 35 zł za godzinę, a w budżecie przeznaczaliśmy 1000 zł na przygotowanie grafiki, mamy ~28,5 godziny pracy grafika.

Jeśli więc pozostałe koszty zostały przez nas oszacowane (albo wydane), kwota ta jest już nie przekraczalna.

Mogłaby się zdarzyć sytuacja, w której grafik spędzi nad pierwszym projektem graficznym 20 godzin - w takim przypadku pozostanie nam jedynie 8 godzin na pozostałe elementy graficzne. Grafik będzie musiał uprościć znacząco obiekty graficzne, żeby w ogóle "zmieścić" się w budżecie.

W takim przypadku będziemy mieli znaczną utratę jakości pomiędzy pierwszym, a kolejnymi ekranami szkolenia.

By takich sytuacji uniknąć, należy ściśle monitorować pracę zespołu.

Dobrym sposobem jest ustalenie reguł. Przykłady to:

Informowanie przełożonych w stałych odstępach czasowych o postępach pracy

Informowanie o opóźnieniach w poszczególnych punktach projektowych

W przypadku problemów technicznych zwrócenie się do eksperta lub przełożonego o pomoc

Zbiór takich reguł może bardzo usprawnić pracę i zabezpieczyć nas przed sytuacją opisaną powyżej.

Tą filozofię warto zastosować do wszystkich dziedzin projektowych. Możemy ocenić ile godzin członek zespołu może spędzić nad konkretnym działaniem, tak by finalnie zmieścić się w budżecie.

Przygotowanie tekstów

Większość tekstów szkolenia przygotowaliśmy w fazie projektowania. Są zawarte w Storyboard oraz w skryptach, które przygotowaliśmy razem z ekspertami. Storyboard przedstawia kolejne ekrany szkolenia, możemy, zatem na ich podstawie przystąpić do implementacji.

W pierwszej kolejności możemy pracować z użyciem edytora tekstu. W związku z tym, że przygotowane przez nas materiały będą użyte przez programistów do scalania ich w kodzie oprogramowanie, ich redakcja powinna odbyć się w formacie RTF albo zwykłym formacie tekstowym.

Pamiętajmy także o dopracowaniu tekstu ze względu na grupę docelową do której go kierujemy. Techniczny język może być zbyt "ciężki" dla sprzedawców, a język marketingowy dla programistów.

Inną ważną kwestią jest podjęcie decyzji, czy zawartość szkolenia będzie jego integralną częścią, czy też będzie pobierana z biblioteki albo bazy danych. Takie rozwiązanie spowoduje, że przetłumaczenie szkolenia na inny język nie będzie stanowiło dużego wyzwania.

W celu uniknięcia kopiowania zawartości kursu, niektórzy projektanci decydują się na zamieszczenie tekstów skonwertowanych do plików graficznych. Pamiętajmy jednak, że taka zmiana może bardzo zwiększyć ciężar naszego szkolenia.

Pisanie kodu oprogramowania

Podczas fazy projektowania badaliśmy dostępne platformy technologiczne pod względem naszego budżetu jak i zdolności naszych programistów. Pamiętajmy też, że zawsze lepiej przygotować aplikację w kodzie, który doskonale znamy, niżeli w języku oprogramowania, który dopiero poznajemy. Zależnie od tego czy zdecydowaliśmy się na aplikację WEB czy też oprogramowanie instalowane z nośnika danych, możemy spotkać się z pewnymi ograniczeniami.

W fazie planowania badaliśmy środowisko pracy naszego użytkownika końcowego. Zależnie od niego, możemy wybrać rodzaj aplikacji.

W Bankach pracownicy często pracują na stacjach roboczych, które mogą nie być wyposażone w czytnik płyt DVD/CD. W małej firmie, możemy spotkać stanowiska pracy, które nie posiadają karty sieciowej ani modemu - a co za tym idzie, brak dostępu do sieci.

Wybór typu WEB czy CD ma znaczenie także ze względu na ciężar aplikacji. CD/DVD pozwala nam na przechowywanie dużych, nie skompresowanych grafik, długich filmów. Jeśli zaś nasza aplikacja ma działać za pośrednictwem internetu, musimy zwrócić uwagę na "ciężar" poszczególnych ekranów szkolenia.

W fazie planowania, podczas badania środowiska pracy użytkownika, znajdziemy odpowiedzi na powyższe zagadnienia. Zbadanie środowiska pracy użytkownika przyda się także przy wyborze platformy programistycznej. W przypadku aplikacji sieciowej, ważne jest aby upewnić się z jaką przeglądarką pracują użytkownicy. W przypadku oprogramowania warto wiedzieć jaki system operacyjny został zainstalowany. Być może środowisko pracy nie jest jednolite. W takim przypadku warto sprawdzić, czy znamy platformę programistyczną która obsłuży wszystkie środowiska.

Podczas pracy programiści natrafiają zapewne na miejsca, w których niewielkie przeróbki mogłyby technicznie znacząco poprawić jakość projektu. Warto jest rozważyć skonsumowanie części czasu przeznaczanego na programowanie w celu dopracowania szczegółu. Inną częstą sytuacją jest natrafienie na funkcjonalność, która według nas może powtórzyć się także w innych aplikacjach. W takim przypadku dobrze by było przygotować kod w postaci biblioteki. To także skonsumuje czas, ale jest wielka szansa, że w przyszłości ta nadmiarowość się nam zrekompensuje.

Bardzo ważną rzeczą podczas tworzenia kodu programowania jest dokumentowanie go. Każda funkcja, biblioteka powinna zawierać przypisy informujące co robi, jak jest wywoływana oraz jaki jest jej oczekiwany efekt. Ponadto całe rozwiązanie powinno zostać opisane w dokumentacji technicznej, zawierającej opisy klas, zależności pomiędzy modułami programistycznymi, język oprogramowania, jego wymagania. W przyszłości wycena stworzonej przez nas aplikacji odbywa się za pomocą "know how" - czyli dokumentacji. Rzadkością jest kupowanie praw do aplikacji bez znajomości i zrozumienia jej kodu.

Tworzenie Grafiki

Przykładowe obiekty oraz projekty graficzne powinny powstać w fazie projektowania. Warto przypomnieć sobie niebezpieczeństwo związane z brakiem kontroli budżetu związanego z obróbką graficzną. Należy zadbać o to, by podczas pracy nad projektem nie doszło do zubożenia obiektów graficznych. Pamiętajmy także, aby graficy wiedzieli do kogo skierowane będzie nasze szkolenie.

W przypadku kursu dotyczącego obsługi drogiej jachtów używanie grafik zapożyczonych z programów typu clip art może doprowadzić do zniechęcenia klienta. Natomiast skupianie się na bardzo dopracowanej i oryginalnej grafice w przypadku szkolenia BHP dla firmy będzie niepotrzebną stratą czasu i pieniędzy.

Przypomnijmy jeszcze dylemat CD czy WEB. Kluczowym problemem w tym sporze staje się właśnie aspekt graficzny. Przygotowanie bardzo bogatej pod względem graficznym prezentacji może - w przypadku gdy ktoś posiada łącze modemowe - doprowadzić do wzrostu frustracji. Pojedynczy ekran szkolenia nie powinien się ściągać dłużej niż dziesięć sekund.

Często możemy się spotkać z niezrozumieniem ze strony zamawiającego. Klient będzie sobie życzył najlepszą jakość grafiki. W takim przypadku należy starać się edukować klienta. Dobrym sposobem jest poświęcenie czasu na przygotowanie przykładu, w którym zaprezentujemy jak wygląda przesłanie przez sieć dużych, słabo skompresowanych plików graficznych - versus - dobrze skompresowane pliki o niższej jakości.

Przypominamy także o sprawdzeniu czy klient ma jakieś wewnętrzne standardy dotyczące kolorów, czcionek, umieszczenia jego logo na każdym ekranie edycji. Możliwym jest, że osoba zamawiająca szkolenie nie zdaje sobie sprawy z ich istnienia. Dla naszego własnego bezpieczeństwa należy poprosić o zatwierdzenie i podpisanie projektu graficznego który przygotowujemy.

Obiekty graficzne możemy przygotować z pomocą programu Ilustrator, Corel Photo painter, PhotoShop Bryce i inne. Wielu grafików swoje pomysły co do obiektów czerpie z natury. Fotografie to także bardzo dobre źródło pozyskania materiału na obiekty.

Klipy Video

Materiały video to silne narzędzie do nauki. W szczególności klipy video przydają się gdy w grę wchodzi emocje, mowa ciała, ludzkie zachowania. Odzwierciedlenie ruchu jest także najlepsze za pośrednictwem filmu.

Jest jednak wiele momentów kiedy wykorzystanie klipu było nie potrzebne. Takie nadużycia są częste w przypadku braku doświadczenia projektanta. O wiele prościej zaprojektować scenkę filmową niż stworzyć hybrydę składającą się z obrazu statycznego i dźwięku połączonego animowanymi przejściami. Często też, wielominutowy klip można by zastąpić lepiej zmontowanym, o wiele tańszym i krótszym. Długie klipy powodują także "rozleniwienie" widza. Materiały video mogą generować najwyższy koszt w całym projekcie. Oszczędność kosztów podczas produkcji klipu video może doprowadzić do powstania produkcji która jakością będzie bardzo odbiegać od tej z telewizji czy kina. Widz szybko zwróci uwagę na różnice, może to też doprowadzić do zaniżenia ogólnej wartości szkolenia - pomimo poniesionych znacznych środków.

Podobnie jak w przypadku pracy nad grafiką, musimy sobie zdawać sprawę ze środowiska pracy użytkownika.

W przypadku aplikacji WEBowych warto zbadać prędkość łącza. Zastanowić się czy umieszczać film w postaci "do ściągnięcia" czy też udostępnić streaming video.

Klipy video możemy przygotować w oprogramowaniu Adobe Premiere, EditDV, Final Cut Pro.

Ścieżki dźwiękowe

Bardzo ważne medium. Korzystanie z narracji wspomaga proces uczenia się osób niedowidzących czy też analfabetów. W przypadku szkoleń językowych (lub szkoleń odbywanych w obcym języku), narrator pozwala lepiej zrozumieć tekst. Osoby słabo rozumiejące dany język słuchając narratora mogą lepiej zrozumieć treść szkolenia.

Narracja jest nie odzowna podczas prezentowania wszelkiego rodzaju prototypów, opisywania animacji, tłumaczenia chartflow. Warto też zwrócić uwagę, że dźwięk połączony ze zmieniającymi się np. zdjęciami pozwoli na otrzymanie efektów zbliżonego do filmu.

W porównaniu do klipów Video - przygotowanie klipów Audio jest o wiele tańsze.

Podczas integracji ścieżek dźwiękowych z naszym produktem warto pamiętać o stworzeniu małego menu które będzie odpowiedzialne za wyłączenie dźwięku, jego regulację oraz możliwość powtórzenia ostatniej sekwencji lub całej lekcji.

Warto pamiętać, że człowiek nie może czytać i słuchać jednocześnie dwóch różnych rzeczy. Dlatego nasze teksty muszą pozostawać w czystej harmonii z tekstem.

Ścieżki dźwiękowe możemy tworzyć używając programów: Forge, SoundEdit, WaveLab.

Wersjonowanie

Podczas pracy nad dużym projektem powinniśmy utrzymywać ciągłą kontrolę nad zespołami nam podległymi. Część z naszych ludzi pracuje nad interfejsem graficznym inni nad ścieżką graficzną, kolejni nad fragmentami oprogramowania. Może się zdarzyć, że zewnętrzna firma przygotowuje dla nas moduł który ma być emulatorem prawdziwej aplikacji. Nasz projekt składa się z wielu "kawałków". Każda wymiana jedno z nich może wpłynąć na stabilność pozostałych. W chwili gdy wszystkie te części połączymy w całość - otrzymamy pierwszą wersję finalnego produktu.

Aby jednak doprowadzić do tego momentu musimy koordynować wymianę fragmentów kodu i programów pomiędzy członkami grupy. W tym celu warto wprowadzić wersjonowanie produktów.

Wersjonowanie to zapisanie wersji oprogramowania z pomocą słownika znaków.

Często programiści mają swoje własne systemy nazewnictwa. Najniebezpieczniejsze jest nadpisywanie poprzednich wersji oraz tak zwane luźne nazewnictwo. NP: "kopia"; "kopia 1"; "dobry"; "stare"; "nowe_2303".

Prawidłowe wersjonowanie powinno zawierać informację o numerze wersji głównej do której dany moduł jest przygotowywany. Numerze kolejnej kompilacji, oraz numerze pożądkowym wersji w których nastąpiły znaczące zmiany (tak zwane kamienie milowe).

Programy, z których pomocą możemy utrzymywać kontrole nad kodem i wersjami oprogramowania są: CVS, Subversion, tortoise(pracujący na platformie CVS).

Przygotowanie dokumentacji (manuale)

Dokumentacja użytkownika

W dokumentacji użytkownika powinniśmy zawrzeć wszystkie niezbędne informacje potrzebne do skutecznego ukończenia szkolenia. Poniższa tabela przedstawia najważniejsze cechy dokumentacji użytkownika.

Strona tytułowa	Powinna informować o nazwie produktu, jego autorach, datę wydania, wersję oprogramowania do którego została napisana.
Tabela zawartości	Tabela powinna zwracać numer wersji dokumentacji, ilość stron, spis treści
Ważne uwagi	Zawarcie uwag dotyczących użytkownika. Bezpieczeństwa nośnika danych na którym oprogramowanie jest nagrane, zasad kopiowania i inne
Wprowadzenie	Powinno zawierać krótki opis produktu, w jakim celu został wyprodukowany, jakie będą zadania uczącego się i inne
Niezbędny sprzęt i oprogramowanie	Techniczne informacje dotyczące komputera oraz platformy systemowej do której program został przygotowany. Może też zawierać informację o dodatkowych pomocach przydatnych do pracy (np. Długopis, kartki, drukarka, skaner)
Instalacja i pierwsze uruchomienie programu	Powinno zawierać informację jak prawidłowo zainstalować oprogramowanie, opisywać komendy które należy wykonać w celu uruchomienia. Informować o planowanym restarcie komputera i inne
Wersja Trial	To możliwość sprawdzenie w szybkim tempie zawartości programu.
Uruchomienie programu	Powinno zawierać proces prawidłowego uruchomienia programu.
Opis kursu	Dokładny opis zawartości kursu. Warto pamiętać, że zbyt dokładne opisanie kursu może wspomagać uczniów podczas końcowego testu
Bibliografia kursu	

Dokumentacja administratora

W przypadku gdy nasz kurs posiada panel administracyjny, opis panelu wraz z funkcjami które z jego pomocą można wykonać powinien znaleźć się w osobnym dokumencie. Warto w nim zamieścić także opis problemów, które może napotkać administrator w swojej pracy.

Dokumentacja techniczna

Przeznaczona dla osób, które mogą potrzebować modyfikować program. Zawiera dokładny opis metody działania programu, algorytmów w nim zastosowanych, rozmieszczenia i sposobu działania poszczególnych komponentów itp. Ze względu na swoją naturę jest ona przeznaczona dla programistów, a dla zwykłego użytkownika właściwie niezrozumiała.

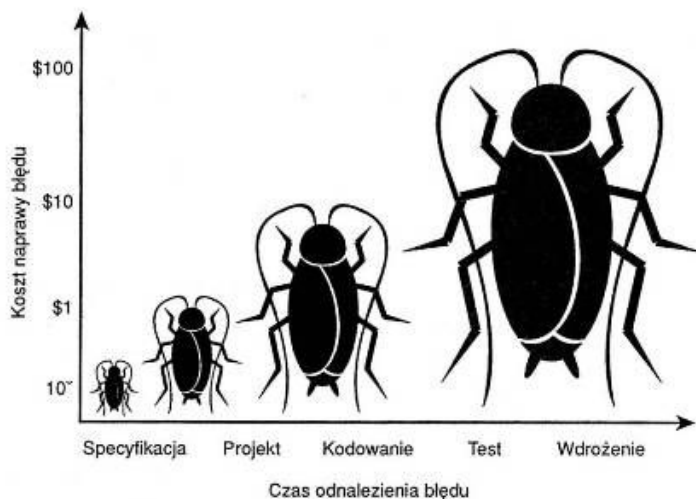
Dokumentacja funkcjonalna

Dokumentacja funkcjonalna powinna opisywać użytkowy aspekt przygotowanego szkolenia. Powinna zawierać opis ekranów edycyjnych, przycisków i innych obiektów. Zadaniem dokumentacji funkcjonalnej jest przedstawienie idei która przyświecała twórcom podczas tworzenia oprogramowania. Dobrze przygotowana dokumentacja funkcjonalna wystarcza do powtórzenia wytworzenia oprogramowania.

Testowanie

Podczas produkcji oprogramowania nieodzownym narzędziem kontrolnym jakości jak i kontekst jest testowanie. Typowe projekty zawierają co najmniej dwa główne etapy testowania. Alfa i Beta test.

Testowanie to czynność która wskazuje nam wszystkie błędy, pomaga kontrolować jakość.



Testowania nie wolno nie doceniać. Co chwile słyszymy o wypadkach firm motoryzacyjnych, komputerowych, programistycznych związanych z brakiem prawidłowego przetestowania produktu. Przykładem może być konferencja Bill'a Gates'a na której pojawił się słynny blue screen. Inne ciekawe przypadki to negatywny test "łosia" podczas sprawdzania Mercedesa Klasy A w roku 1997. Jak się okazało winnym był błąd komputera sterującego systemem amortyzatorów. Innym jeszcze bardziej bolesnym błędem była katastrofa podczas lądowania na Marsie Lądownika NASA. Przyczyną okazało się nieprzewidziane ustawienie wartości 1 bitu!!!

Testowania wersji alfa

To inaczej testowanie wewnętrzne - bez współudziału klienta. Mogą w nim brać udział wszystkie osoby zaangażowane w projekt. Większość pracy podczas testowania wykonuje tester. Testowanie polega na wielokrotnym badaniu wszystkich funkcji które zostały zaimplementowane w projekcie. W tym celu może zostać utworzony dokument: scenariusz testów. Dokument ten zawierać będzie wszystkie "ścieżki" przejść. Test alfa jest jednym z oficjalnych procesów testowania. Warto jednak pamiętać, że testowanie oprogramowania powinno odbywać się cały czas podczas fazy produkcji. (wraz z postęпами prac).

Testowanie to także proces sprawdzania jak zachowa się program w chwili nie typowego zachowanie się użytkownika. Testerzy za zwyczaj sprawdzają co stanie się gdy wywołają funkcje wielokrotnie, klikną "wstecz" w chwili zapisywania wyników testów, zbadają wszystkie obszary ekranu na których teoretycznie nie da się wywołać akcji. Ich zadaniem jest "złamać" program, doprowadzić do jego zawieszania lub nie prawidłowego działania.

Po zakończonym procesie testowania należy przeanalizować otrzymane informacje.

Rewizja wersji

Mając już wyniki testów możemy przystąpić do eliminacji błędów. Ważne jest by podczas poprawiania oprogramowania nie doprowadzić do stworzenia nowych błędów. Po każdej rewizji musi nastąpić ponowna - nawet krótka faza testowania wewnętrznego. Po wszystkich zmianach należy także dopracować dokumentację. Szczególnie pod koniec pracy nad projektem, gdy czas jest bardzo mało, zapomina się o tym ważnym elemencie.

Testowanie wersji beta

To inaczej testowanie końcowe, z udziałem klienta. Prawdopodobnie podczas pracy nad oprogramowaniem klient wielokrotnie brał udział w akceptacjach poszczególnych rozwiązań. Jednak oficjalnym sprawdzeniem działania oprogramowania jest właśnie ten moment. Podczas testów beta klient często wyznacza grupę testerów których zadaniem, podobnie jak wcześniej naszych pracowników, jest sprawdzenie stabilności aplikacji.

Jeśli klient nie ma własnego procesu testowania, warto zaproponować mu następujące rozwiązanie:

- 1 Wyselekcjonowanie grupy uczniów
- 2 Wytlumaczenie procedury testowania
- 3 Dowiedzenie się jaką wiedzę dotyczącą materiału szkoleniowego posiadają
- 4 Obserwuj ich podczas pracy ze szkoleniem
- 5 Przeanalizuj i zapisz ich wnioski po szkoleniu
- 6 Sprawdź ich wiedzę po przeprowadzonym szkoleniu
- 7 Przeanalizuj program

Rewizja końcowa

Po zakończeniu testowania oprogramowania z klientem należy podjąć decyzję, czy oprogramowanie w obecnej formie jest gotowe do przekazania go użytkownikom końcowym. To może być bardzo trudna decyzja, gdyż najczęściej podczas fazy beta odnajdziemy jakieś błędy które wcześniej nie były przez nas dostrzeżone. Logiczne jest więc, że im więcej testów wewnętrznych alfa przeprowadzimy, tym większa szansa, że testowanie beta będzie sukcesem.

Przekazanie produktu klientowi - zakończenie projektu

Jeśli podczas poszczególnych faz produkcji oprogramowania nawiązaliśmy współpracę z klientem. Dbaliśmy by zawsze był dobrze poinformowany, oraz by brał udział w najważniejszych momentach w projekcie, nie powinniśmy mieć kłopotu z dopełnieniem formalności jaką jest podpisanie przekazania produktu. Jeśli podczas beta testów doszło do pewnych błędów albo problemów warto zaproponować rozwiązanie ich w kolejnych wersjach oprogramowania. (dotyczy to tylko błędów małych, błędy krytyczne powodują ponowne przeprowadzanie testów alfa oraz beta). Zakończenie projektu nie oznacza zakończenia pracy z klientem. To w jaki sposób zakończymy współpracę może w przyszłości wpłynąć na nasz wizerunek i dobre imię.

Jeśli podczas trwania projektu klient wprowadzał znaczące zmiany w projekcie, warto o tym pamiętać. Jeśli wcześniej ustaliliśmy określone dodatkowe pieniądze z tego tytułu, podczas oddania projektu warto mieć przygotowane pozostałe rozliczenia.

Zdarza się, że usługodawca chcąc utrzymać dobre relacje z klientem rezygnuje z dodatkowego honorarium.

Streszczenie modułu

Niniejszy dokument przybliża sposób prowadzenia projektu eLearningowego, opisując fazę produkcji projektu eLearningowego.

Słownik kluczowych pojęć

Zarządzenie Projektem	<p>Zarządzanie projektem jest to zbiór czynności wykonywanych w celu osiągnięcia wyznaczonych celów. Zawiera się w nim planowanie, harmonogramowanie i utrzymywanie postępów w czynnościach składających się na projekt. Najprościej można także powiedzieć, iż jest to nauka o utrzymywaniu ryzyka porażki na możliwie niskim poziomie w całym cyklu życia projektu. Ryzyko to bierze się głównie z niepewności związanej z przyszłymi wydarzeniami na każdym etapie przedsięwzięcia.</p> <p>Z innego punktu widzenia, zarządzanie projektem można zdefiniować jako naukę o definiowaniu i osiąganiu celów przy jednoczesnej optymalizacji użycia zasobów (np. czasu, pieniędzy, ludzi, itd.).</p> <p>Zarządzanie projektem to pole działania i odpowiedzialności jednej osoby, zwanej menadżerem projektu. Taka osoba rzadko uczestniczy bezpośrednio w czynnościach, które prowadzi do doprowadzenia projektu do celu, lecz raczej służy jako koordynator, którego zadaniem jest utrzymanie postępów i współpracy pomiędzy członkami projektu w taki sposób, by zredukować całkowite ryzyko porażki.</p> <p>W języku polskim przyjęły się głównie dwie nazwy określające tę samą dziedzinę wiedzy, zarządzanie projektami (zamiast zarządzania projektem) oraz zarządzanie przedsięwzięciami, z czego to pierwsze cieszy się największą popularnością. Wśród osób zajmujących się tą nauką zawodowo dużym powodzeniem cieszy się także angielskojęzyczne sformułowanie Project Management.</p>
Standard	<p>Wspólnie ustalone kryterium, które określa powszechne, zwykle najbardziej pożądane cechy czegoś, np. wytwarzanego przedmiotu (np. standardem jest, że każdy współcześnie wytwarzany telewizor wyświetla kolory) czy ludzkiego zachowania (norma kulturowa). Standard to czasem także podstawowa, najprostsza wersja produktu.</p> <p>W ekonomii zejście poniżej standardu zwykle dyskwalifikuje dany wytwór i powoduje brak popytu na ów przedmiot, lub obniżenie jego ceny. Przekraczanie standardów jest zwykle mile widziane u konsumentów, lecz wiąże się ono z większymi kosztami produkcji i sprzedaży.</p> <p>Standard w tym ujęciu jest kryterium, którego nie można zmierzyć (można zmierzyć pewne cechy kryterium, jak np. dopuszczalna długość telefonu komórkowego itp., ale samego standardu dokładnie zmierzyć się nie da), jest płynna granica ustalana na bieżąco zarówno przez grupy konsumenckie, jak i osobnych odbiorców.</p> <p>W technice standard to zestaw parametrów, zwykle posiadający nazwę (np. PAL w telewizji), który zapewnia odpowiedni poziom jakości, bezpieczeństwa, wygody lub zgodności z innymi wytworami techniki. To ostatnie jest ważne np. w kolejnictwie (rozmiary szyn), produkcji śrub (rozmiary gwintów) czy takich dziedzinach techniki jak informatyka i telekomunikacja, zwłaszcza w sieciach komputerowych (np. w Internecie), ze względu na łatwość i szybkość zmian zachowań komputerów, związana z nieporównywalnie niższymi kosztami niż w innych dziedzinach przemysłu (np. zmiana systemu rozmiarów torów kolejowych w Europie w porównaniu z wprowadzeniem protokołu IPv6), i konieczność utrzymywania komunikacji między dużą ilością różnych rozwiązań.</p> <p>Opracowywaniem standardów zajmują się często odpowiednie organizacje, np. ISO, IEEE, World Wide Web Consortium (W3C) czy JPEG. Niektóre normy narodowe stają się faktycznym standardem międzynarodowym w danej dziedzinie, np. amerykańskie normy ANSI czy niemieckie DIN. W Polsce istnieje zestaw regulacji o nazwie Polska Norma (PN).</p>
Organizacja wirtualna	<p>Nowy typ organizacji charakterystyczny dla wielkich korporacji charakteryzujący się tym, że nie posiadają one własnych środków produkcji w postaci zakładów produkcyjnych czy nawet pracowników, natomiast wynajmują zewnętrznych podwykonawców, którzy realizują cele organizacji.</p> <p>Zazwyczaj są to organizacje dysponujące znacznym kapitałem i mające już duże uznanie na rynku. Jej działanie ogranicza się do nadawania produktom swojej marki, przy czym proces produkcji nie jest przez tego typu organizację kontrolowany. Charakterystyczna dla tego typu organizacji jest elastyczność przy wyborze podwykonawców w skali globalnej i stosunkowo łatwa delokalizacja kapitału. Zazwyczaj dla potaniaenia kosztów produkcji</p>

	organizacje tego typu poszukują coraz tańszej siły roboczej w różnych regionach świata, bardzo często lokując w końcu fabryki w krajach Trzeciego Świata. Naomi Klein w swojej pozycji No Logo ostro skrytykowała tego typu organizacje, między innymi firmy Reebok, Nike, czy Microsoft za brak odpowiedzialności za warunki pracy oferowanych przez podwykonawców wynajmowanych przez te firmy.
Planowanie	Faza projektu w której badamy własne zasoby, przygotowujemy ofertę, nakreślamy budżet.
Klient	Osoba lub organizacja dla której świadczona jest usługa. Klient wewnętrzny - to osoba lub organizacja zamawiająca usługę znajdująca się w tej samej firmie. W przypadku gdy oprogramowanie tworzymy dla siebie, stajemy się klientem wewnętrznym. Klient jest osobą opłacającą projekt.
Użytkownik	Osoba lub grupa osób, które finalnie będą użytkowały oprogramowanie. Użytkownikiem szkolenia eLearningowego może być wyspecjalizowana grupa ludzi, lub też bliżej nie określona. Np. segment rynku lub "wszyscy studenci"
Nauka w trybie mieszanym (<i>blended-learning</i>)	Ten terminy zamiennie opisują podejście do kształcenia, w którym łączy się metodę w klasie i metodę na odległość, w której instruktor lub tutor spotyka się ze swoimi uczniami (albo metodą twarzą w twarz, albo przy użyciu środków technicznych), a baza zasobów zawierająca materiały i ćwiczenia jest udostępniana uczniom. Dodatkowo mogą być wykorzystywane niektóre z metod eLearning.
System Zarządzania Uceniem się (<i>Learning Management System</i>)	Zestaw narzędzi eLearningowych dostępnych poprzez wspólny administrowany interfejs. System zarządzania uczeniem się może być traktowany jako platforma na której zbiera się i używa kursów on-line i komponentów kursów.
Storyboard	Scenorys (ang. storyboard), scenopis obrazkowy - seria obrazów i szkiców, będących wskazówkami przy filmowaniu dla reżyserów, scenografów, operatorów, aktorów i montażyстів.
Komunikacja synchroniczna	Komunikacja w czasie rzeczywistym (np. chat).
Flowchart	Schemat blokowy (ang. block diagram, flowchart) - diagram, na którym procedura, system albo program komputerowy są reprezentowane przez opisane figury geometryczne, połączone liniami zgodnie z kolejnością wykonywania czynności wynikających z przyjętego algorytmu rozwiązania zadania. Schemat blokowy pozwala dostrzec istotne etapy algorytmu i logiczne zależności między nimi. Zależnie od przedstawianego stosowane są różne zestawy figur geometrycznych zwanych blokami, których kształty reprezentują umownie rodzaje elementów składowych.
Produkcja	Implementacja (z łac.), informatyczny proces przekształcania abstrakcyjnego opisu systemu lub programu na obiekt fizyczny: komputer lub działający program zapisany w konkretnym języku programowania; także obiekt fizyczny będący efektem takiego przekształcenia, np. implementacja systemu operacyjnego lub kompilatora dla konkretnego typu komputera.
Prototyp	urządzenie, obwód lub program zaprojektowany i zbudowany w celu zademonstrowania zdolności do budowy urządzenia docelowego. Podczas budowy prototypu inżynierowie wprowadzają pierwszy raz w życie swoje nowe pomysły. Jeżeli uda się zbudować prototyp i on działa, to można przystąpić do budowy finalnego urządzenia. Często podczas projektowania niektóre rzeczywiste zjawiska związane z konstrukcją zostają pominięte. Prototyp pozwala na odkrycie ich nieznanego wpływu i wprowadzenie korekt. Jeżeli pierwszy prototyp nie jest udany, to buduje się kolejne, aż do uzyskania w urządzeniu spełniającego założenia. Prototyp może fizycznie bardzo różnić się od ostatecznego urządzenia, jeżeli chcemy sprawdzić tylko pewną jego cechę.
Burza mózgów	technika wywodząca się z psychologii społecznej, która ma na celu doskonalenie decyzji grupowych. Burza mózgów jest również formą dyskusji dydaktycznej, wykorzystywaną jako jedna z metod nauczania. Zalicza się ją wówczas do metod aktywizujących, która stanowi podgrupę metod problemowych.
Beta Tester	to osoba, która przed wydaniem oprogramowania komputerowego testuje jego jakość, wydajność oraz stabilność na wersji beta. Pomaga przedsiębiorstwom komputerowym w tworzeniu lepszego oprogramowania poprzez składanie raportów. Bycie beta testerem jest bardzo często traktowane jako swoista nobilitacja (szczególnie w przypadku gier komputerowych) ze względu na niewielki rozmiar grupy, która ma dostęp do oprogramowania przed oficjalną premierą. W przypadku oprogramowania Open Source beta testowanie ma charakter działalności wolontariackiej - każda wersja programu jest dostępna dla wszystkich użytkowników, a zainteresowani mogą zgłaszać błędy w trakcie użytkowania.

Komunikacja asynchroniczna	Komunikacja w czasie opóźnionym (np. email).
----------------------------	----------------------------------------------

Literatura podstawowa i poszerzająca

Portal Wikipedia - www.wikipedia.pl

Alessi Stephen M., Trollip Stanley R. (2001) "Multimedia for Learning - Methods and Development"

Rosenberg Marc J. (2001) "E-Learning - Building successful online learning in Your organization - strategies for delivering knowledge in the digital age"

Schank, Roger C. (2002) "Designing World-Class E-learning"

Kitaoka Vance, Shea Andrea, Uehara Randal (1998) "Creating a Storyboard for Video Production"
www2.hawaii.edu/~ricky/etec/storyboarding.html

Oprogramowanie do tworzenia Storyboard
<http://www.atomiclearning.com/storyboardpro>

♦ Jak usprawnić proces testowania oprogramowania: procedury, metodologia i narzędzia ♦
www.erudis.pl

Kruse Kevin "Creating Scripts and Storyboards for e-Learning"
http://www.e-learningguru.com/articles/art2_5.htm

Pytania problemowe

Co jest konieczne do wyceny wartości produktu finalnego?
