

Zespołowy projekt CASE, etap określenia wymagań

Inżynieria wymagań opisuje schemat pozyskiwania, dokumentowania i zarządzania wymaganiami. Wdrożenie inżynierii wymagań w ramach przedsięwzięcia informatycznego ma na celu zminimalizowanie negatywnych konsekwencji niedoskonałości wymagań stawianych oprogramowaniu. Stosowanie inżynierii wymagań w znacznym stopniu przybliży zespół projektowy do wytworzenia oprogramowania zaspokajającego rzeczywiste potrzeby klienta, na czas i w ramach wyznaczonego budżetu. Etap określenia wymagań częściowo pokrywa się z etapem analizy, tak że w tym samym czasie zespół powinien rozpocząć prace analityczne.

Cel etapu

Opracować specyfikację wymagań systemowych (SWS).

Zadania do wykonania

- Opracować raport dla Zleceniodawcy:
 - o Zdefiniować cele przedsięwzięcia.
 - o Opisać zakres przedsięwzięcia.
 - o Opisać systemy zewnętrzne, z którymi system będzie współpracować.
 - o Przygotować szkicowy opis: wymagań, modelu systemu; proponowanych rozwiązań.
 - o Oszacować koszty projektu.
 - o Dołączyć wstępny harmonogram prac.
- Opracować pełną wersję specyfikacji wymagań (Software Requirement Specification, SRS) zgodną z zaleceniami IEEE 830-1998:
 - o Opracować słownik projektu.
 - o Przygotować listę wymagań funkcjonalnych i нефункциональных.
 - o Zidentyfikować aktorów.
 - o Przeprowadzić identyfikację przypadków użycia.
 - o Opracować diagramy przypadków użycia.
- Opracować styl firmowy projektu:
 - o Logo (logotyp, symbol), tagline (slogan), wizytówkę;
 - o Papier firmowy;
 - o Szablon firmowy do opracowania dokumentacji projektowej.

Wymagana dokumentacja etapu

1. Raport dla Zleceniodawcy.

2. Specyfikacja wymagań systemowych (może podlegać edytowaniu na kolejnych etapach).
3. Opis stylu firmowego projektu

Źródła wiedzy

1. Wojciechowski A., *Wprowadzenie do inżynierii wymagań*, http://www.inmost.org.pl/articles/Wprowadzenie_do_inzynierii_wymagaA
2. Ksawery Skowron, Poziomy celów przypadków użycia, eX Blog. - <http://kskowron.enzo.pl/2006/10/08/poziomy-celow-przypadkow-uzycia.html>
3. Z cyklu: "USE CASE'owe historie" - za czy przeciw? - <http://forum.4programmers.net/viewtopic.php?id=97099>
4. Narysuj mi UMLa - <http://mimas.ceti.pl/tarpit/archiwum/1088408766>
5. Mandar Chitnis, Pravin Tiwari, Lakshmi Ananthamurthy, Creating Use Case Diagrams - <http://www.developer.com/design/article.php/2109801/>
6. Scott W. Ambler, UML 2 Use Case Diagrams - <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>
7. Sun Developer Network: Tutorials. UML Modeling: Creating Use Case Diagrams - http://developers.sun.com/prodtech/javatools/jsenterprise/learning/tutorials/jse8/uml_use_case.html

Porady, wskazówki i podpowiedzi

- Przy opracowaniu słownika projektu wymagana jest precyzyjna i jednoznaczna definicja najbardziej istotnych dla projektantów pojęć dziedzinowych.
- Wymagania funkcjonalne warto podzielić na funkcje główne i dodatkowe, wśród wymagań нефункциональных warto wyróżnić wymagania uzupełniające i ograniczenia.
- Przy identyfikowaniu aktorów pod uwagę należy brać użytkowników systemu, osoby potrzebne do utrzymania systemu (np. administrator) oraz systemy współpracujące. W identyfikacji aktorów może pomóc słownik projektu. W opisie aktora powinna się znajdować jego rola i przypadki użycia aplikacji, z których korzysta.
- W identyfikacji przypadków użycia mogą pomóc słownik projektowy i lista wymagań funkcjonalnych. Opis przypadku użycia powinien zawierać: numer, nazwę, cel, scenariusz podstawowy, scenariusze alternatywne, priorytet, wymagania нефункциональные, akcję inicjującą.
- Do opracowania diagramu przypadków użycia można użyć łatwego do opanowania edytora Violet. Alternatywą może być użycie wolnych narzędzi UML jak StarUML czy ArgoUML.

Dodatkowe informacje dotyczące fazy określania wymagań

Istnieją różne poziomy określania wymagań; najbardziej istotne są wymagania funkcjonalne, obrazują one hierarchię funkcji konstruowanego systemu, bardzo pomocne

w wizualizacji wymagań są diagramy przypadków użycia; istotne mogą być również wymagania niefunkcjonalne. Podstawowe rezultaty fazy określania wymagań to dokumenty ustaleń wspólnych klienta i wykonawcy.

Na tym etapie należy określić, jakie wymagania musi spełniać projektowany program. Najważniejsze, by w tej fazie nie wdawać się w szczegóły implementacyjne. Takie myślenie dla programisty może być trudne. Aby to zrobić, trzeba spróbować poczuć się jak przyszły użytkownik, wyobrazić sobie swój program jako już skończony. Użytkownik będzie jedynie widzieć interfejs graficzny programu, jego okna, poprzez które będzie starać się uzyskać pożądaną dla siebie funkcjonalność. Jakie elementy powinien zawierać przyszły interfejsu użytkownika? Jaki powinien być wachlarz opcji, ogólnie i w szczegółach, żeby spełniał oczekiwania użytkownika? To są pytania, na które trzeba szukać odpowiedzi w tej fazie.

Jedynym wyjątkiem od tej zasady, by nie myśleć o samym kodowaniu, jest kontrolowanie cały czas, czy wymyślane oczekiwania nie są nierealne do zrealizowania. Stawiać należy przede wszystkim na takie rozwiązania, które są jednocześnie proste do kodowania, a przynoszą bardzo dobry efekt. Nie warto zapominać też rozpatrzeć, które funkcje są najważniejsze dla użytkownika, a z których można zrezygnować albo je ograniczyć.

Zawartość dokumentu specyfikacji wymagań według normy ANSI/IEEE „Recommended Practice for Software Requirements Specifications”:

1. Wstęp
 - 1.1. Cel
 - 1.2. Zakres
 - 1.3. Definicje, akronimy i skróty
 - 1.4. Referencje, odsyłacze do innych dokumentów
 - 1.5. Krótki przegląd
2. Ogólny opis
 - 2.1. Walory użytkowe i przydatność projektowanego systemu
 - 2.2. Ogólne możliwości projektowanego systemu
 - 2.3. Ogólne ograniczenia
 - 2.4. Charakterystyka użytkowników
 - 2.5. Środowisko operacyjne
 - 2.6. Założenia i zależności
3. Specyficzne wymagania
 - 3.1. Wymagania funkcjonalne (funkcje systemu)
 - 3.2. Wymagania niefunkcjonalne (ograniczenia).