

Gromadzenie wymagań z użyciem szablonu VOLERE

Ewa Magiera, Aldona Galas, Magdalena Turzańska
Uniwersytet Śląski w Katowicach
e-mail: magiera@us.edu.pl, alga@adm.us.edu.pl, matu@adm.us.edu.pl

Abstrakt. Autorki są członkami powołanego w Uniwersytecie Śląskim zespołu do zaprojektowania i implementacji systemu obsługi dziekanatów z wykorzystaniem narzędzi i bazy Oracle. Do procesu gromadzenia wymagań różnego typu użyto opracowanego przez Suzanne i James'a Robertson szablonu Volere. W referacie zostaną przedstawione zalety i wady zastosowanej metodologii.

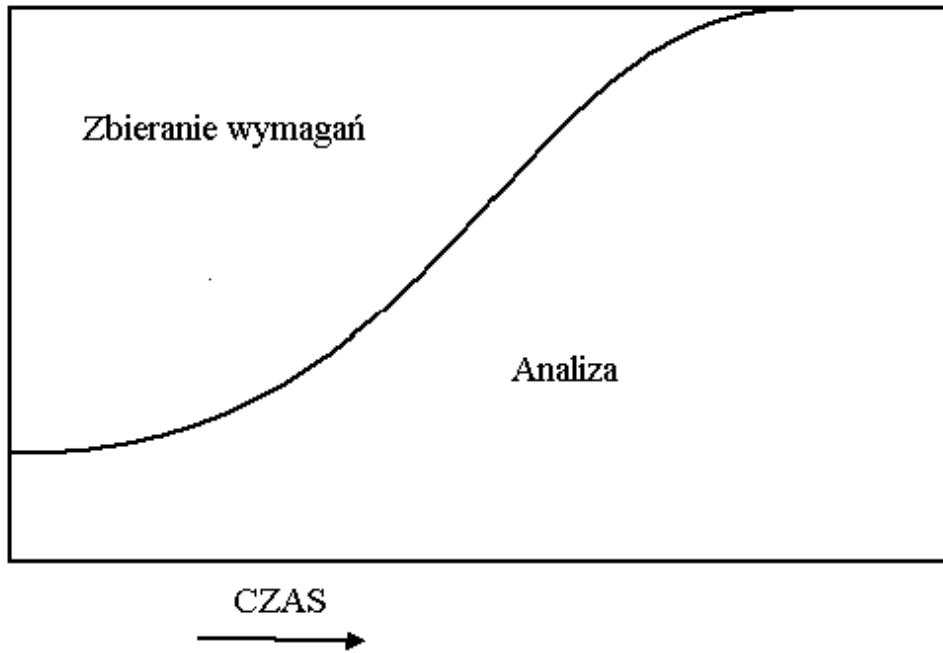
1. Wstęp

Jednym z ważniejszych etapów każdego projektu informatycznego jest etap gromadzenia, weryfikacji oraz opracowania wymagań, które decydują o późniejszym sukcesie zrealizowanego projektu. Wszystkie wymagania powinny zostać zebrane przed pracami projektowymi i implementacyjnymi. Odkrywanie nowych cech produktu w trakcie realizacji projektu powoduje wzrost kosztów i wydatnie zmniejsza efektywność realizowanych zadań. Etap zbierania wymagań istotny jest nie tylko w przypadku tworzenia nowych projektów informatycznych, ale również w przypadku kiedy zostaje podjęta decyzja o modernizacji i rozbudowie działających rozwiązań informatycznych, jak również przy zakupie gotowego oprogramowania. Bez względu na fakt czy projekt dotyczy problemu realizowanego dla własnych potrzeb czy na zlecenie dla klienta, ściśle i pełne opisanie wymagań zapewni sprawne działanie opracowywanego produktu oraz zwiększy efektywność pracy grupy realizującej projekt.

Istnieje wiele podejść do technik gromadzenia wymagań, można wykorzystać metodologię związaną z używanym oprogramowaniem, którego będziemy używać do realizacji projektu, może to być metodologia zupełnie z oprogramowaniem niezwiązana np. język modelowania danymi (UML) czy notacja przepływu danych T.DeMarco. Nie jest celem artykułu deprecjonowanie istniejących metod wykorzystywanych w celu zebrania wymagań, natomiast chcemy zaprezentować metodologię nazwaną przez jej twórców Suzanne i James'a Robertsonów - Volere i będącą rezultatem ich długoletniej pracy i doświadczenia w zakresie analizy systemowej. Jest ona zupełnie niezależna od oprogramowania używanego do wykonania projektu. Volere został wykorzystany na etapie gromadzenia wymagań w projekcie prowadzonym w Uniwersytecie Śląskim i mającym na celu zbudowanie oprogramowania wspomagającego proces dydaktyczny.

2. Proces zbierania wymagań w realizacji projektu

Proces zbierania wymagań rozpoczyna analizę systemową, im większa wiedza o realizowanym produkcie tym zespół realizujący mniej czasu poświęca na zbieranie wymagań a więcej na analizę. Relację w czasie pomiędzy tymi dwoma działaniami przedstawia rys. 1.

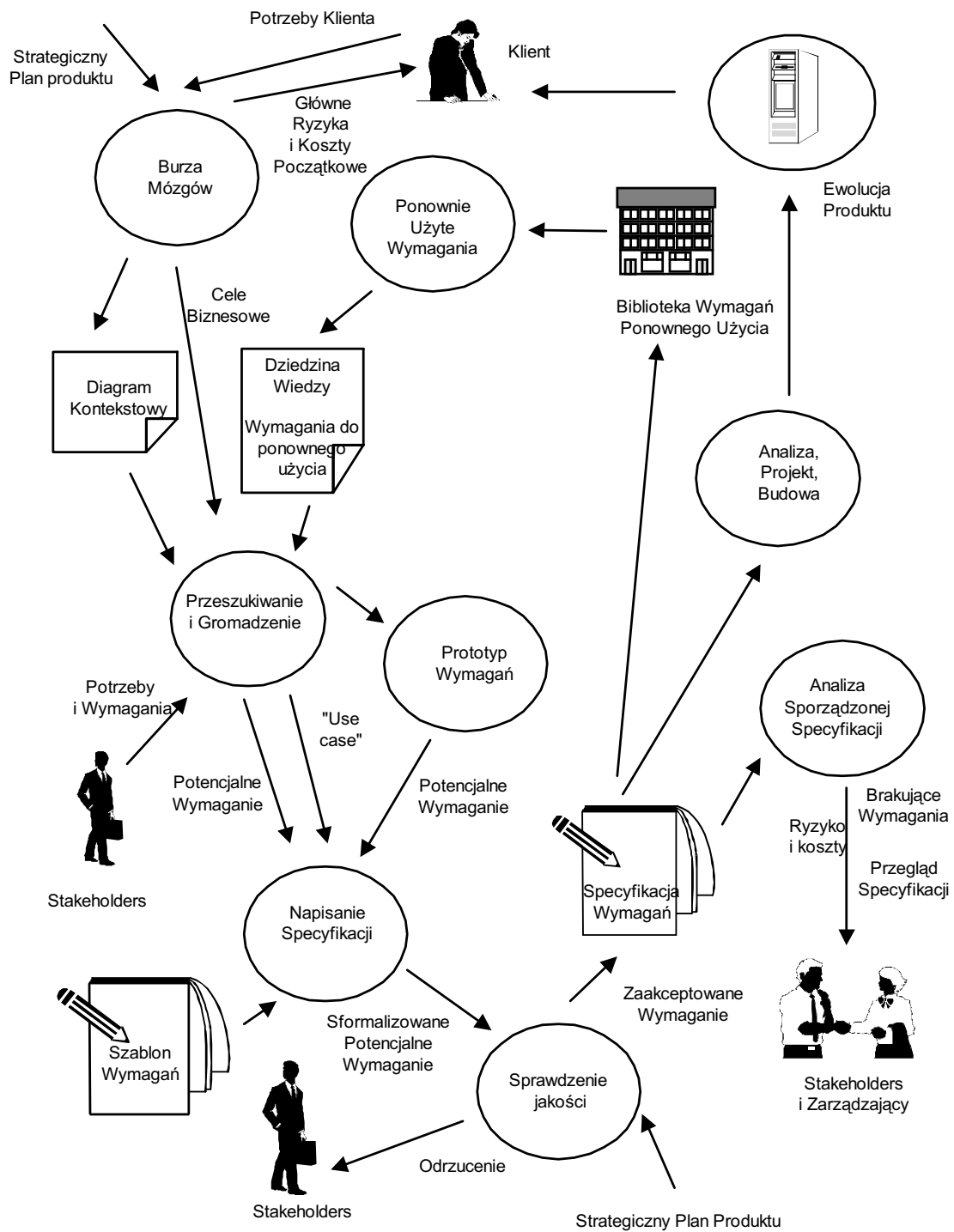


Rys. 1. Relacje pomiędzy zbieraniem wymagań a analizą w czasie

Pod pojęciem wymagań rozumiemy wszystko to co produkt po zakończeniu realizacji wszystkich etapów projektu powinien robić, jakie winny być jego właściwości pod względem efektywności, jakości, wyglądu itp. Wyróżnić można następujące trzy podstawowe grupy wymagań:

- funkcjonalne,
- нефункционалне,
- ograniczenia.

Sam proces gromadzenia wymagań jest złożony i składa się z różnych czynności, często powtarzających się, co zostało zaprezentowane na rysunku 2.



Rys. 2. Proces zbierania wymagań

3. Opis metodologii Volere

Wszystkie wymagania jak również ograniczenia winny znaleźć się w specyfikacji wymagań, tak aby stanowiły kompletny opis produktu. W celu przygotowania pełnej specyfikacji S. i J. Robertson opracowali szablon pod nazwą Volere, składający się z następujących 26 punktów:

1. Przyczyny podjęcia realizacji projektu oraz cele jakie powinno się po zakończeniu prac uzyskać. W punkcie tym należy dokładnie opisać dlaczego projekt jest realizowany, jakie korzyści będą efektem jego zakończenia dla końcowego użytkownika. Korzyści, wynikające z wdrożenia produktu winny zostać przedstawione w sposób „mierzalny” (np. zwiększenie ilości studentów obsługiwanych w dziekanacie).
2. Określenie następujących osób (jednostek): zlecającej realizację, ewentualnych nabywców oraz wszystkie osoby mające jakikolwiek wpływ na przebieg realizacji projektu i postać końcową produktu (stakeholders). Należy nazwać wszystkich wspomnianych wyżej, jako że to oni decydują o budżecie, o akceptacji lub odrzuceniu wyniku końcowego jak również są źródłem wszelkiej informacji o realizowanym produkcie. W wypadku wykonywania oprogramowania na własne potrzeby zamawiający i kupujący to ta sama jednostka lub osoba (np. uczelnia).
3. Użytkownicy – lista potencjalnych użytkowników produktu. Dla każdej kategorii użytkowników należy przedstawić następujące informacje:
 - Nazwa użytkownika – może to być nazwa grupy zawodowej lub osoby (np. inżynierowie produkcji, sekretarki, itp.).
 - Rola użytkownika – odpowiedzialność zawodowa wynikająca z zakresu obowiązków (np. przygotowanie list stypendialnych).
 - Poziom wiedzy i doświadczenia jakie posiada użytkownik w stosunku do produktu (np. umiejętność obsługi aplikacji o zbliżonych właściwościach).
 - Poziom znajomości technologii wykorzystywanej w projekcie (np. administratorzy sieci).
 - Inne cechy użytkownika, które mogą mieć znaczący wpływ dla projektu.
 - Określenie ważności użytkownika (przydzielenie priorytetów) w procesie zbierania wymagań:
 - kluczowi użytkownicy- posiadający największe doświadczenie i wiedzę – najważniejsi w procesie gromadzenia wymagań (np. prodziekani ds. studenckich, kierownicy dziekanatów),
 - użytkownicy drugiego poziomu – będą produkt używać, ale w wypadku konfliktu co do wymagań pomiędzy nimi a kluczowymi użytkownikami – zdanie kluczowych będzie preferowane (np. pracownicy dziekanatów),
 - użytkownicy trzeciego poziomu – rzadko używający produktu, słabo wykwalifikowani, nie wnoszący do projektu wiele nowego.
 - Określenie personaliów osób wchodzących do danej grupy użytkowników, należy określić jakiego rodzaju wiedzy oczekujemy od danego użytkownika, wskazane jest aby dokonać oszacowania ilości czasu, który użytkownik musi spędzić z osobą zbierającą wymagania tak aby można było uzyskać jak najwięcej koniecznych informacji.
4. Ograniczenia – ta część szablonu dotyczy ograniczeń jakie należy nałożyć na produkt zarówno w zakresie jego walorów funkcjonalnych jak i estetycznych.
 - Ograniczenia dotyczące rozwiązań technologicznych używanych do realizacji produktu, które mogą decydować o środowisku w jakim produkt ma pracować lub wprost będą stanowić jego część.
 - Ograniczenia wynikające z zakresu (granic) działania produktu.
 - Ograniczenia wynikające z rodzaju aplikacji współpracujących, ale nie będących częścią produktu finalnego.
 - Identyfikacja komercyjnych produktów, które ewentualnie mogłyby z finalnym produktem współpracować.
 - Ograniczenia wynikające z miejsca, w którym produkt będzie fizycznie funkcjonował.

- Ograniczenia czasowe - termin ostateczny realizacji produktu.
 - Ograniczenia wynikające z wysokości budżetu, który może zostać przeznaczony na realizację projektu.
5. Nazewnictwo i definicje – należy w specyfikacji wymagań umieścić jednoznaczne definicje używanych pojęć, nazw i skrótów. Da to gwarancje, iż wszystkie osoby biorące udział w projekcie używając tych samych określeń mówią o tych samych rzeczach i zjawiskach.
 6. Fakty zewnętrzne mogące mieć związek z projektem i produktem.
 7. Przypuszczenia – hipotezy prawdopodobnych zdarzeń lub faktów, które nie są prawdziwe w momencie tworzenia specyfikacji ale mogą zdarzyć się w przyszłości i mieć wpływ na realizowany produkt. Celem tej sekcji jest uświadomienie wszystkim uczestnikom projektu faktów prawdopodobnych, mogących zdecydować o sukcesie lub porażce.
 8. Zasięg projektu.
 - Diagram (ogólny) przedstawiający całość realizowanego projektu.
 - Lista modułów (części) na które można podzielić realizowany projekt i może być ona punktem wyjścia do gromadzenia bardziej szczegółowych wymagań.
 - Granice produktu – określenie dokładnych granic i zasięgu działania produktu.
 9. Wymagania funkcjonalne i wymagania danych.
 - Wszystkie wymagania funkcjonalne czyli co produkt powinien robić.
 - Wymagania danych – pierwsze przedstawienie danych dotyczących realizowanego projektu. Może być przedstawiony w postaci diagramu związków encji, obiektowego modelu, UML lub inaczej.
 10. Wymagania dotyczące wyglądu i estetyki użytkowej produktu, wynikające z potrzeb potencjalnych klientów.
 11. Wymagania dotyczące funkcjonalności użytkowej produktu, ze względu na grupę przyszłych użytkowników (opisują np. łatwość korzystania z produktu, szybkość opanowania obsługi, itp.).
 12. Wymagania dotyczące szybkości, dokładności pracy, trybu (np. on lub off-line), trwałości, pojemności czyli wymagań wynikających z ewentualnych ograniczeń technologicznych.
 13. Wymagania dotyczące warunków pracy produktu, wynikające z otaczającego środowiska pracy.
 14. Wymagania dotyczące utrzymania produktu oraz opisanie różnych środowisk, w których produkt musi pracować.
 15. Wymagania dotyczące bezpieczeństwa i warunków w jakich to bezpieczeństwo może zostać zachowane (np. zróżnicowane uprawnienia użytkowników do danych).
 16. Wymagania kulturowe i polityczne.
 17. Wymagania wynikające z obowiązującego prawa (np. regulamin studiów, przepisy dotyczące ochrony danych osobowych).
 18. Wymagania „otwarte” takie, które zostały zauważone, nie można jednak na obecnym etapie sformułować obowiązujących wniosków końcowych.
 19. Przegląd gotowych oferowanych na rynku produktów, które ewentualnie mogłyby być brane pod uwagę.
 20. Nowe problemy – opis sposobu w jaki nowy produkt będzie oddziaływać na środowisku, w którym będzie pracować, w punkcie tym można również przedstawić czego produkt nie powinien robić.
 21. Opis zadań – w punkcie tym należy przedstawić szacunkowo czas niezbędny do realizacji poszczególnych etapów i zadań projektu, stosowaną metodykę, nie zapominając o konwersji danych i szkoleniu użytkowników końcowych.

22. Wymagania odnoszące się do przystosowania (konwersji) istniejących danych, procedur, itp., tak aby mogły pracować w nowym środowisku.
23. Ryzyko – przedstawienie wszystkich możliwych zagrożeń, które mogą zdecydować o niepowodzeniu projektu.
24. Kosztorys.
25. Dokumentacja użytkownika, która będzie częścią budowanego produktu – określenie treści i zakresu.
26. Wymagania czekające – nie dotyczą aktualnego projektu, lecz mogą się pojawić w przypadku budowy kolejnej wersji produktu.

W przeprowadzonym w Uniwersytecie Śląskim procesie zbierania wymagań, każde wymaganie zostało opisane przy pomocy przedstawionej na rysunku nr 4 karty wypełnionej przykładowymi danymi.

Tabela 1. Praktyczne użycie szablonu Volere

Nr_wymagania:	1	Typ_wymagania:	9	Moduł:	1
Opis	Podstawowe dane osobowe studenta w systemie: <ul style="list-style-type: none"> - PESEL, - NIP, - nazwisko, - nazwisko rodowe, - imię 1, - imię 2, - imię ojca, - imię matki, - data urodzenia, - miejsce urodzenia, - adres stały, - adres tymczasowy, - adres do korespondencji, - płeć, - narodowość, - obywatelstwo, - nr i seria dowodu osobistego, - kod Kasy Chorych, - nazwa Banku i konto bankowe, - e-mail, - rok immatrykulacji. 				
Uzasadnienie:	Informacja niezbędna o studencie. Pesel lub NIP może być identyfikatorem studenta.				
Źródło:	Stary system informatyczny, wymagania jednostek.				
Kryteria:	Dane gromadzone w dziekanatach na podstawie dokumentów przedłożonych przez studenta.				
Zależności:					
Sprzeczność:					
Potwierdzenie:	Regulamin studiów.				

Daty:	24.01.2001,
-------	-------------

4. Podsumowanie

Autorzy szablonu w 26 punktach przedstawili wszystkie aspekty i problemy związane z realizacją projektu informatycznego, co wpływa na uporządkowanie i kompletność gromadzonych wymagań. Nawet w wypadku naszego projektu można mówić o pewnej nadmiarowości. Nie wszystkie punkty od 1 do 26 zostały uwzględnione, jako że realizowany projekt nie ma charakteru komercyjnego. Jednak nawet w tej sytuacji, wykonanie zebrania wymagań funkcjonalnych systemu z użyciem szablonu Volere ma wiele zalet, np.:

- użycie języka naturalnego powoduje, iż wymagania są zrozumiałe dla wszystkich uczestników projektu, począwszy od analityków a skończywszy na użytkownikach końcowych,
- użycie zunifikowanych formularzy dla każdego wymagania gwarantuje przejrzystość i zrozumiałość zapisu,
- gromadzić i zapisywać wymagania mogą osoby o różnym poziomie wiedzy informatycznej,
- metoda jest niezależna od softwaru i hardware,
- metoda jest efektem długoletniej pracy i doświadczenia praktycznego twórców.

Bibliografia

1. S.Robertson, J,Robertson: Mastering the requirements process, Addison-Wesley, 1999, ISBN 0-201-360462
2. <http://www.atlsysguild.com/>