

KATEDRA INFORMATYKI STOSOWANEJ POLITECHNIKI ŁÓDZKIEJ

WPROWADZENIE DO ZASAD POZYSKIWANIA WYMAGAŃ

Opracowanie: mgr inż. Radosław Adamus¹

¹ Opracowano na podstawie materiałów edukacyjnych Software Engineering Institute (SEI) - "Lecture Notes on Requirements Elicitation" autorstwa Sridhar Raghavan, Gregory Zelesnik, i Gary Ford, numer dokumentu CMU/SEI-94-EM-10, copyright 1994 by Carnegie Mellon University. Zagwarantowana możliwość wykorzystania w celach niekomercyjnych.

Spis Treści

Spis Treści	2
1. Wprowadzenie: Opowieść o trzech studentach.....	3
2. Proces pozyskiwania wymagań.....	4
2.1 Terminologia	4
2.2 Ogólna procedura wydobywania wymagań	5
2.3 Uczestnicy procesu pozyskiwania wymagań	5
3. Wyniki procesu pozyskiwania wymagań.....	6
3.1 Efekty dobrego procesu.....	6
3.2 Efekty złego procesu	7
4. Najważniejsze problemy związane z procesem pozyskiwania wymagań.....	7
4.1 Artykulacja potrzeb	8
4.2 Bariery komunikacyjne	9
4.3 Wiedza i ograniczenia poznawcze	9
4.4 Kwestie techniczne.....	10
5. Przegląd technik pozyskiwania wymagań.....	11
5.1 Techniki wysokopoziomowe.....	12
5.1.1 Joint Application Design (JAD).....	12
5.1.2 Adaptive Loop Framework (ALF)	13
5.1.3 Prototypowanie.....	13
5.2 Techniki szczegółowe	14
5.2.1 Burza mózgów (<i>ang. Brainstorming</i>).....	14
5.2.2 Wywiady	14
5.2.3 PIECES.....	14
6. Podsumowanie	15

1. Wprowadzenie: Opowieść o trzech studentach

Pewnego razu było sobie trzech studentów informatyki: Pat, Terry i Chris. Na zajęciach z programowania prowadzący zadał następujące ćwiczenie do wykonania:

Napisz program, który wczyta 100 liczb naturalnych, posortuje je w porządku rosnącym, wyświetli listę tak posortowanych liczb wraz z obliczoną wartością średnią.

To, co usłyszeli studenci było zbiorem *wymagań*, jakie oprogramowanie musi spełniać. Nasi studenci nie mieli większych problemów z rozwiązaniem powyższego zadania. Pat i Chris zaczęli od napisania na kartce papieru algorytmu i szkieletu kodu programu. Terry od razu podszedł do klawiatury i bezpośrednio rozpoczął pisanie programu.

Teraz nasi studenci, a tak naprawdę świeżo upieczeni absolwenci z tytułem magistra informatyki rozpoczęli swoją pierwszą pracę. Pat znalazł zatrudnienie w dużej firmie przemysłowej. Pewnego dnia Pat oraz pozostali pracownicy wydziału inżynierii oprogramowania, na specjalnie zwołanym zebraniu, otrzymali, od wiceprezesa do spraw sprzedaży i marketingu następujące zadanie:

Stwórzcie zautomatyzowany system, który pozwoli nam przetwarzać zamówienia przynajmniej 24 godziny szybciej niż obecna średnia oraz dostarczać nasz produkty przynajmniej trzy dni wcześniej niż obecnie.

Terry znalazł zatrudnienie w firmie zajmującej się tworzeniem systemów informatycznych dla potrzeb lotnictwa, a w obecnej chwili dla nowego samolotu pasażerskiego Z-686. Zespół właśnie otrzymał następujące zadanie:

Stwórzcie oprogramowanie, dla Z-686, które pozwoli mu lądować na większości lotnisk bez pomocy pilota.

Chris rozpoczął pracę dla firmy specjalizującej się w tworzeniu oprogramowania dla komputerów osobistych. Szef firmy wezwał wszystkich nowych inżynierów oprogramowania i zlecił im następujące zadanie.

Stwórzcie oprogramowanie, które sprzeda się przynajmniej w milionie kopii i którego detaliczna cena będzie wynosiła przynajmniej 200\$.

W przeciwieństwie do sytuacji z zajęć programowania, żaden z naszych znajomych po usłyszeniu polecenia, nie skierował się do klawiatury komputera. Zdawali sobie sprawę, że potrzebują dużo więcej informacji o tym, co tak naprawdę planowane oprogramowanie musi robić.

W jaki sposób mogą się tego dowiedzieć? Odpowiedzią na to pytanie są techniki pozyskiwania wymagań.

Aby zrozumieć proces pozyskiwania wymagań, spójrzmy początkowo na niego z ogólnego punktu widzenia. Jaka terminologia jest wykorzystywana, kto uczestniczy w procesie wydobywania wymagań, jakie są podstawowe procedury wykorzystywane w procesie? Przystudiujemy i porównamy wyniki poprawnych i błędnych procesów wydobywania wymagań. Następnie przedyskutujemy najważniejsze problemy tego procesu. Na końcu naszkicujemy zasady kilku podstawowych technik wydobywania wymagań wykorzystywanych w inżynierii oprogramowania.

2. Proces pozyskiwania wymagań

Pozyskiwanie wymagań jest jednym z najbardziej krytycznych kroków w projekcie informatycznym. Ponad 30-letnie doświadczenie Inżynierii Oprogramowania pokazało, że niepoprawne, niekompletne, bądź źle zrozumiane wymagania są najważniejszą przyczyną projektów niskiej jakości, przekroczeń budżetu oraz opóźnień w dostarczaniu oprogramowania do klienta. Z tego względu, umiejętność wdrażania systematycznego procesu pozyskiwania wymagań jest fundamentalną wiedzą, jaką musi posiadać dobry inżynier oprogramowania.

2.1 Terminologia

Wiele terminów służy do opisywania procesu poznawania i rozumienia wymagań dla systemu informatycznego. *Inżynieria Wymagań* jest ogólnym pojęciem obejmującym wszystkie czynności związane z wymaganiami. Przede wszystkim na inżynierię wymagań składają się następujące procesy:

Pozyskiwanie wymagań – proces, służący odkrywaniu, artykułowaniu, przedstawianiu wymagań klienta i użytkownika względem systemu.

Analiza wymagań – proces wnioskowania o pozyskanych wymaganiach. W procesie tym zawierają się takie działania jak: wyszukiwanie konfliktów i niespójności w wymaganiach, łączenie powiązanych wymagań czy identyfikowanie wymagań brakujących.

Specyfikacja wymagań – proces transformowania wymagań do jednej z następujących reprezentacji:

- języka naturalnego,
- języka formalnego,
- języka symbolicznego,
- języka graficznego.

Na końcu tego procesu powstaje dokument – specyfikacja wymagań.

Walidacja wymagań – proces, w którym klient lub użytkownik oprogramowania potwierdza poprawność, kompletność oraz ważność specyfikacji. Proces ten może podnieść dokument specyfikacji do rangi prawnego porozumienia między stronami.

W rzeczywistości, te cztery powyższe procesy nie występują niezależnie, i niemożliwe jest sekwencyjnie wykonywane. Można powiedzieć, że zlewają się ze sobą w jeden iteracyjnie wykonywany proces. Na przykład przedstawienie wymagania w formalnym, graficznym zapisie pomaga w identyfikowaniu konfliktów bądź braków w wymaganiach. Natomiast proces walidacji często powoduje wydobycie nowych wymagań bądź szczegółów, które nie zostały uprzednio zidentyfikowane przez użytkownika.

Należy zwrócić uwagę na to, że termin „pozyskiwanie” (ang. elicitation) wymagań nie jest ogólnie przyjęty. Często zamiennie stosuje się takie pojęcia jak: identyfikacja (ang. indentifying), gromadzenie (ang. gathering), określanie (ang. determining), formułowanie (ang. formulating), lub ujawnianie (ang. exposing, extracting). Terminy te wzajemnie się uzupełniają i dopełniają znaczenie pojęcia. Dla przykładu *gromadzenie* sugeruje, że

wymagania już gdzieś się znajdują, a naszym zadaniem jest zebranie ich razem, *formułowanie* zwraca uwagę na potrzebę wytworzenia wymagań, natomiast *ujawnianie* daje do zrozumienia, że wymagania zostały w pewien sposób ukryte przez użytkownika. Patrząc ogólnie na proces pozyskiwania wymagań, każde z tych pojęć niesie w sobie część prawdy, składającą się na całość znaczenia.

2.2 Ogólna procedura wydobywania wymagań

Zdecydowanie najpopularniejszym rodzajem wydobywania wymagań jest pobieranie ich bezpośrednio od użytkowników systemu. W takim wypadku proces wydobywania może zostać opisany w bardzo ogólnej terminologii przy wykorzystaniu 5 następujących kroków.

1. Identyfikacja istotnych źródeł wymagań użytkowników.
2. Zadanie odpowiednich pytań (odpowiednim osobom) w celu zrozumienia potrzeb użytkowników.
3. Przeanalizowanie zebranych informacji w celu wykrycia błędów, niespójności czy braków.
4. Potwierdzenie zrozumienia wymagań przez użytkowników.
5. Stworzenie odpowiedniego dokumentu opisującego wymagania.

Techniki wydobywania wymagań rozwijają te kroki poprzez definiowanie szczegółów procesu, określanie pytań lub kategorii pytań, które powinny zostać zadane, formalne określanie przebiegu spotkania, określanie sposobu zachowania osób i grup osób, definiowanie szablonów zapisu pozyskanej informacji. Wybrane techniki zostaną przedstawione w pkt. 5.

2.3 Uczestnicy procesu pozyskiwania wymagań

W wysiłek zdobycia wymagań jest zazwyczaj zaangażowana grupa ludzi. Jedną z nich jest inżynier oprogramowania (czasem wręcz specjalista zwany inżynierem wymagań), odpowiedzialny za wytworzenie specyfikacji wymagań, kierujący całym procesem. Jest on często wspierany przez innych inżynierów oprogramowania, specjalistów od tworzenia dokumentacji lub pracowników biurowych.

W proces jest również zaangażowany potencjalny użytkownik oprogramowania. W typowym systemie informatycznym, np. takim, z jakim miał do czynienia Pat, jest wiele osób, które będą miały bezpośredni kontakt z systemem: przedstawiciele handlowi, personel zajmujący się dokonywaniem zamówień, personel oddziału dostaw, czy pracownicy księgowości. Użytkownikami, w tym wypadku mogą być również menedżerowie i kadra kierownicza, a szczególnie ci z nich, którzy przeprowadzają proces autoryzacji budowanego systemu.

Terry (pracuje w firmie tworzącej oprogramowanie dla przemysłu lotniczego) ma do czynienia z trzema rodzajami użytkowników: Inżynierami projektującymi samolot, którzy znają zasady budowy i działania jego systemów oraz sposób, w jaki oprogramowanie musi z tymi systemami współpracować. Inna grupa użytkowników to osoby, które są w stanie powiedzieć, jakie konkretnie zadania oprogramowanie musi wykonywać. Ze względu na to, że kwestie dotyczące lotnictwa muszą być zatwierdzone, przez odpowiednie instytucje państwowe, w trakcie pozyskiwania wymagań może być potrzebna osoba reprezentująca taki urząd. Również piloci powinni być zaangażowani w proces (np. przy określaniu wymagań dotyczących interfejsu użytkownika).

Chris napotka zupełnie inne problemy. Jeżeli nowy pakiet oprogramowania będzie nową wersją istniejącego (np. „nowy i udoskonalony” procesor tekstowy) to dobrze, jeżeli w procesie pozyskiwania wymagań uczestniczy reprezentatywna grupa użytkowników istniejącej wersji. Mogą oni odpowiedzieć na pytania, co im się podoba lub nie podoba w istniejącym pakiecie, jakie nowe funkcje powinny się pojawić itp. Z drugiej strony, jeśli nowy pakiet oprogramowania ma być unikatowy na rynku, dużo trudniej jest określić wymagania. Badając rynek można zidentyfikować jego potrzeby a co za tym idzie ogólne wymagania. Jednak szczegółowe wymagania mogą pochodzić dopiero z prototypów i wyników testów przeprowadzanych przez użytkowników.

Z powyższych rozważań można wysnuć jedną ważną tezę – nie ma pojedynczej osoby, która wiedziałaby wszystko o planowanym oprogramowaniu. W dobrze przeprowadzonym procesie pozyskiwania wymagań jest zazwyczaj wielu współuczestników.

3. Wyniki procesu pozyskiwania wymagań

Namacalnym rezultatem procesu pozyskiwania wymagań jest zbiór wymagań, który może być wykorzystany przez zespół projektowy do wytworzenia oprogramowania. Jednak istnieje wiele innych, mniej namacalnych wyników tego procesu, które mogą mieć wpływ na sukces całego projektu. Jakość tych wyników różni się ze względu na jakość samego procesu pozyskiwania wymagań.

3.1 Efekty dobrego procesu.

Klienci i/lub użytkownicy oprogramowania często przystępują do procesu pozyskiwania wymagań, mając tylko ogólne pojęcie o sposobie działania systemu i z bardzo nie wielką wiedzą o zaletach i możliwościach technologii informatycznej. Dobry proces wydobywania wymagań pomaga im odkryć i zrozumieć własne oczekiwania, szczególnie podczas oddzielania tego, co *chcieliby mieć* od tego, czego *rzeczywiście potrzebują*. Współpraca z inżynierem oprogramowania może również pomóc im w zrozumieniu ograniczeń wynikających z technologii, regulacji prawnych itp. Dodatkowo poznają możliwe alternatywy technologiczne i proceduralne, które mogą być rozważane przy produkcji systemu.

Podsumowując klienci i użytkownicy partycypując w tym procesie mają możliwość poznania i zrozumienia wpływu swoich decyzji dotyczących wymagań na sam proces wydobywania wymagań. Rezultatem tego jest mniejsza liczba niemiłych niespodzianek, na jakie narażeni są klienci otrzymujący gotowy produkt. Dodatkowo użytkownicy mają szanse współdzielić z inżynierami oprogramowania wizję tworzonej aplikacji, rozważając możliwe do zrealizowania rozwiązania. Dzięki temu uzyskują poczucie satysfakcji z prowadzonych działań, czują się doinformowani, wierzą, że ryzyko, jakie podejmują jest minimalizowane i są przekonani o sukcesie przedsięwzięcia.

Podobnie ma się rzecz z twórcami oprogramowania uczestniczącymi w procesie pozyskiwania wymagań. Mają oni poczucie, że rozwiązują rzeczywiste problemy użytkowników, co jest jednym z najważniejszych warunków sukcesu projektu. Wspólna dyskusja nad wymaganiami daje im przeświadczenie, że system, który zostanie wytworzony, będzie lubiany przez użytkowników, efektywny i niepowodujący ujemnych skutków ubocznych. Oczywiście otrzymują również jasno sprecyzowaną, wysokopoziomową specyfikację projektowanego systemu.

Twórcy oprogramowania poznają dziedzinę, dla której projektują oprogramowanie, zbierają wiele dodatkowych i pomocniczych informacji, które mogą przydać się w późniejszej fazie rozwoju oprogramowania.

3.2 Efekty złego procesu

Najpoważniejszym wynikiem złego procesu pozyskiwania wymagań jest to, że twórcy oprogramowania rozwiązują nieprawidłowy problem. Jest to gwarancją niepowodzenia projektu.

Nawet, jeżeli uda się doprowadzić do sytuacji, kiedy tworzone oprogramowanie rozwiązuje prawdziwy problem, to zły proces pozyskiwania wymagań może prowadzić do wielu negatywnych efektów. Klienci i użytkownicy mogą być niezadowoleni: taka sytuacja zdarza się często, gdy projektanci oprogramowania tak naprawdę wcale ich nie słuchają lub grają w procesie rolę dominującą, forsując swoją własną wizję systemu, interpretując na swój sposób to, co słyszą od klientów i potencjalnych użytkowników. Rezultatem niezadowolenia jest gorsza współpraca, pozyskane informacje są niekompletne, co wpływa negatywnie na proces wytwarzania i opóźnia czas dostarczenia oprogramowania.

Zły proces pozyskiwania wymagań często prowadzi do realizacji chaotycznego projektu. Twórcy mogą w pewnym momencie odkryć, że brakuje im istotnych informacji, co powoduje konieczność kolejnych spotkań z klientami i użytkownikami. Projektanci, ze względu na brak zrozumienia potrzeb użytkowników, mogą podjąć błędne decyzje. W takich sytuacjach wymagania mogą zmieniać się dużo częściej, co powoduje zwiększenie nakładów na zarządzanie konfiguracją, jest przyczyną opóźnień, marnowania zasobów projektu i w ostateczności może skończyć się jego anulowaniem.

Wszystko to może spowodować utratę pieniędzy, reputacji firmy oraz obniżyć morale twórców oprogramowania.

4. Najważniejsze problemy związane z procesem pozyskiwania wymagań.

Pozyskiwanie wymagań jest trudnym i mało precyzyjnym procesem. Aby przeprowadzić go z pozytywnym efektem, należy zdawać sobie sprawę z najważniejszych trudności, z jakimi można się spotkać w jego trakcie.

Poprzez nasze wcześniejsze rozważania przewijał się termin użytkownika oprogramowania. Przez to pojęcie rozumieliśmy tak osobę, która system wykorzystuje w pracy (jeżeli bierzemy pod uwagę człowieka, a nie inny system), jak i osobę, która za oprogramowanie płaci i/lub je zamawia (klient). Na przykład w firmie, w której pracuje Pat, użytkownikami oprogramowania są sprzedawcy oraz pracownicy biura przetwarzającego zamówienia. W przypadku Terry'ego użytkownikami mogą być piloci i pasażerowie samolotu, natomiast klientami są inżynierowie projektujący urządzenia kontrolujące lot samolotu. Chris ma do czynienia jeszcze z innym przykładem użytkowników. Są nimi osoby, które kupią pakiet oprogramowania, natomiast klientami są osoby, które znają potrzeby użytkowników (np. pracownicy działu badań rynkowych).

4.1 Artykulacja potrzeb

Pierwszą klasę problemów stanowią te, które są powiązane z artykulacją potrzeb użytkowników. W ich skład wchodzi problemy użytkowników związane z wyrażeniem ich potrzeb, jak i problemy twórców oprogramowania z ich zrozumieniem (tak samych użytkowników jak i ich potrzeb).

1. Użytkownicy planowanego systemu mogą zdawać sobie sprawę ze swoich potrzeb, jednak nie są w stanie poprawnie ich wyrazić. Można to porównać z sytuacją, w której zdajesz sobie sprawę z tego, że jesteś głodny i idziesz do restauracji. Jednak będąc już tam nie możesz się zdecydować, co zjesz albo nie możesz zrozumieć menu. Napotkałeś na problemy z wyartykułowaniem swoich wymagań. Powiedzenie kelnerowi „Jestem głodny” jest sposobem wyrażenia swoich potrzeb, jednak nie wystarczy, aby mógł on na nie odpowiednio zareagować.
2. Użytkownicy mogą nie zdawać sobie sprawy ze swoich potrzeb. Mogą nie rozumieć, w jaki sposób technologia mogłaby im pomóc. Na przykład przedstawiciele handlowi pracujący w firmie Pat'a mogą nie wiedzieć, że dysponując komputerem przenośnym wyposażonym w modem mogą dokonywać zamówień bez potrzeby wizyty w firmie.
3. Użytkownicy mogą być świadomi swoich potrzeb, ale mogą obawiać się ich wyartykułować. Na przykład nowo zatrudniony przedstawiciel handlowy może mieć problemy z zapamiętaniem kodowych oznaczeń wszystkich artykułów i chciałby, aby system przechowywał je dla niego w łatwo dostępnym miejscu. Jednak wiedząc, że inni pracownicy takich problemów nie mają, więc może zostać posądzony o niekompetencje, nie wypowiada swojej prośby.
4. Użytkownicy i twórcy napotykają problemy związane z różnym rozumieniem tych samych pojęć i terminów. Wyrazy, takie jak *system* czy *integracja* są szeroko stosowane jednak ich znaczenie może być inne, gdy są wypowiedziane przez osoby z różnych specjalności.
5. Użytkownicy mają problem z podjęciem decyzji dotyczącej sposobu rozwiązania konkretnego zagadnienia. Jest to wynikiem tak braku rozumienia konsekwencji swoich decyzji jak i brakiem wiedzy na temat alternatywnych możliwości.
6. Żadna z osób nie posiada całościowej wizji systemu. Bez względu na to jak dobrze użytkownik wyraża swoje potrzeby, inni użytkownicy mogą mieć inne bądź dodatkowe wymagania dotyczące oprogramowania lub inne priorytety dla tych samych kwestii. Taka sytuacja często zdarza się przy bardzo złożonych systemach, na które pojedynczy użytkownik może patrzeć z bardzo ograniczonej, jemu specyficznej perspektywy. Na przykład użytkownik systemu przetwarzania tekstu mógł nigdy nie korzystać z automatycznych spisów treści i z tego powodu wśród jego wymagań taka potrzeba nie pojawi się.
7. Twórcy systemu mogą tak naprawdę wcale nie słuchać użytkowników, nie brać pod uwagę wszystkich szczegółowych potrzeb wyrażanych przez nich. Taka sytuacja zdarza się, gdy twórcy systemu mają poczucie, że wszystko już zrozumieli lub, gdy zaczęli patrzeć na system z punktu widzenia konkretnego modelu projektowego i implementacyjnego.
8. Twórcy systemu nie potrafią spojrzeć na system oczami użytkowników. Niemożliwe jest wtedy uchwycenie prawidłowego kontekstu, w którym system ma działać.
9. Twórcy oprogramowania stosują taktykę „wyższości” swej pozycji w stosunku do pozycji użytkowników systemu. Sprawiają zazwyczaj wtedy wrażenie znawców problemu. Użytkownicy czują się wtedy niepewnie i nie są w stanie wyartykułować swoich potrzeb.

4.2 Bariery komunikacyjne

Wiele problemów z pozyskiwaniem wymagań związanych jest z różnicami w sposobie komunikowania się użytkowników z twórcami.

1. Użytkownicy i twórcy pochodzą „z innych światów”, posługują się różnymi słownikami pojęć. Użytkownicy mogą być finansistami, lotnikami, przedsiębiorcami. Twórcy systemu są informatykami.
Na przykład Terry spotka się z sytuacją, że gdy zaczyna mówić o hierarchiach klas oraz spójności modułu, użytkownicy zaczynają patrzeć na niego jak na przybysza z innej planety. On sam też ma takie odczucie, gdy użytkownicy częstują go terminami typu „promienie VOR”, czy „interferencje RF”.
2. Użytkownicy mają na uwadze inne aspekty systemu; wysokopoziomowe atrybuty – funkcjonalność, wiarygodność, natomiast twórcy patrzą na system z niższego poziomu – wymagane zasoby, użyte algorytmy, wymagania sprzętowe.
3. Problemy związane z formą lub narzędziem komunikacji. Język naturalny jest bardzo niejednoznaczny. Jest to jego zaletą w codziennej komunikacji. Jednak przy wyrażaniu wymagań jest bardzo dużą niewygodą. Można, więc postawić pytanie, dlaczego używamy języka potocznego w rozmowach z użytkownikami? Odpowiedź jest bardzo prosta – jest to najczęściej jedyny możliwy sposób komunikacji.
4. Pozyskiwane wymagania jest z natury procesem socjologicznym. Ludzie, którzy w nim uczestniczą różnią się od siebie. Jedni są stanowczy, inni ulegli, jedni lubią mówić o szczegółach, inni wolą pojęcia bardziej abstrakcyjne. Powoduje to specyficzną „niekompatybilność” komunikacyjną. Osoba kierująca procesem musi wykrywać te problemy i próbować regulować sposób komunikacji.
5. Ludzie są różni: mają różne typy osobowości, różne systemy wartości. Może to prowadzić do niespodziewanych problemów z porozumiewaniem się. Przypadek taki zdarzył się podczas projektowania systemu dla pewnego uniwersytetu. Kierownikiem projektu była osoba wysoko postawiona w hierarchii firmy – wykonawcy. Z tego względu chciał on rozmawiać z osobami, które miały porównywalne stanowisko – dziekanami oraz prorektorem. Pozostali członkowie zespołu projektowego rozmawiali tylko z pracownikami niższego szczebla – sekretariatów, dziekanatów.

4.3 Wiedza i ograniczenia poznawcze

Klienci, użytkownicy, twórcy systemu są tylko ludźmi i wnoszą w proces określoną wiedzę, którą posiadają oraz ograniczenia poznawcze, którym podlegają. Jest to czynnik zmienny zależny od konkretnej osoby.

1. Osoba pozyskująca wymagania musi posiadać odpowiedni poziom wiedzy z zakresu dziedziny projektowanego systemu. Częstym błędem jest sytuacja, w której zespół złożony z użytkowników oraz twórców nie ma wystarczającej wiedzy na temat projektowanego systemu, co w rezultacie prowadzi do błędnych decyzji projektowych.
2. Pamięć ludzka jest niedoskonała. Użytkownicy i twórcy mogą zapomnieć, co zostało powiedziane i ustalone. Co więcej, można inaczej zinterpretować to co zostało zapisane niż to, co zostało powiedziane. Nawet, jeżeli jesteśmy pewni, że zapisaliśmy wszystkie niezbędne informacje, może się zdarzyć, że później, podczas interpretacji popełnimy błędy.
3. Mimo, iż będziemy starać się wyrażać pozyskane wymagania ilościowo i statystycznie, takie dane mogą być potem zinterpretowane niezgodnie z naszymi

zamiarami. Wynika to z tego, że ludzie bywają tendencyjni powołując się na własne doświadczenie.

4. Wraz ze wzrostem stopnia skomplikowania problemu, ludzie mają tendencję do uproszczeń lub pomijania niektórych elementów. Powoduje to zniekształcenie naszego punktu widzenia na określone zagadnienie.
5. Często z góry zakładamy sposób rozwiązania problemu (ponieważ taki jest nam znany), i tak problem naginamy, aby pasował do naszego rozwiązania.
6. Niektórzy ludzie zawężają problem do kilku aspektów, zazwyczaj tych, które najlepiej rozumieją, bądź tych, które, według nich, najlepiej dany problem charakteryzują.
7. W przypadku dużych systemów musi zazwyczaj powstać wiele próbnych sformułowań problemu, aby w końcu móc wypracować ostateczną jego wersję. Wielu ludziom nie starcza cierpliwości na tak dokładne badania.

4.4 Kwestie techniczne

Istnieje duża grupa problemów związanych z pozyskiwaniem wymagań, które możemy określić jako „techniczne”. Wszystkie powinny być wzięte pod uwagę w trakcie procesu. Do najważniejszych należą:

1. Złożoność problemów, które są rozwiązywane przy wykorzystaniu technik informatycznych systematycznie rośnie. To powoduje, że wiedza o dziedzinie, do której będzie należał projektowany system, musi być coraz bardziej szczegółowa. Dodatkowo pojawia się problematyka wpływu, jaki może wywierać tworzony system na jego otoczenie (tak systemowe, jaki i społeczne). Może się okazać w trakcie procesu pozyskiwania wymagań, że ani twórcy, ani użytkownicy nie są przygotowani na rozwiązywanie aż tak złożonych problemów.
2. Wymagania zmieniają się wraz z upływem czasu. Proces pozyskiwania wymagań powoduje zwiększanie się wiedzy użytkowników na temat systemu. Może to powodować, że wcześniejsze ustalenia tracą ważność. Należy być bardzo ostrożnym, aby nie okazało się, że wraz z końcem procesu pozyskiwania wymagań posiadamy duży zbiór potrzeb użytkowników, które są już nieaktualne.
3. Technologie informatyczne bardzo szybko się zmieniają. Wymagania, które jeszcze wczoraj były nie do spełnienia, dzisiaj mogą być już osiągalne.
4. Istnieje wiele źródeł wymagań. Użytkownicy systemu nie koniecznie muszą być tym najlepszym. Takimi lepszymi źródłami mogą okazać się administratorzy, personel pomocniczy. Kadra kierownicza zazwyczaj posiada informację o tym, w jaki sposób pewne zadania powinny być wykonywane, jakim ograniczeniom muszą podlegać. Mogą dodatkowo istnieć pewne regulacje prawne, czy standardy, które system musi spełniać.
5. Naturalna nowatorskość systemu nakłada dodatkowe ograniczenia na proces pozyskiwania wymagań. Jeżeli system jest podobny do systemów, które zostały wcześniej zbudowane, może czerpać z wymagań, które zostały zebrane dla jego poprzedników oraz informacji zwrotnych, które można uzyskać od użytkowników istniejących już systemów. Jeżeli system jest absolutną nowością, wymaga dużo bardziej solidnego i czasochłonnego procesu pozyskiwania wymagań.

W procesie zdobywania wymagań dla systemu budowanego dla określonych potrzeb oraz dla konkretnego klienta można założyć, że jest niekwestionowanym autorytetem, jeżeli chodzi o potrzeby użytkowników. Natomiast, jeżeli system ma być oferowany również innym klientom, twórcy powinni również wziąć pod uwagę

konkurencyjne rozwiązania oraz wymagania, które stawiają inni zainteresowani produktem.

W przypadku komercyjnych pakietów oprogramowania osobistego proces pozyskiwania wymagań jest mocno uzależniony od badań rynkowych, możliwości, jakie posiadają produkty konkurencyjne i wyników wstępnej komunikacji z potencjalnymi użytkownikami. System, który jest rozwijany od wielu lat, tworzone są nowe jego wersje, powinien w procesie pozyskiwania wymagań uwzględniać informacje o błędach pojawiających się w kolejnych wersjach oraz potrzebach klientów związanych z potencjalnymi udoskonaleniami.

W przypadku systemów czasu rzeczywistego, pozyskiwanie wymagań wiąże się z decyzjami związanymi z tym, jaka część funkcji będzie wykonywana przez dedykowany sprzęt, a jaka przez oprogramowanie.

5. Przegląd technik pozyskiwania wymagań

Techniki pozyskiwania wymagań, które są wykorzystywane w inżynierii oprogramowania powstały w celu rozwiązania jednego bądź grupy problemów opisanych w poprzednim rozdziale. Niektóre dedykowane są do rozwiązywania problemów komunikacyjnych, inne technicznych. Niektóre są opisane na stosunkowo wysokim poziomie abstrakcji, określając ramy, w jakich powinien zmieścić się proces pozyskiwania wymagań, inne bardziej szczegółowo opisują taktykę, jaką należy stosować, aby pozyskać wymagania dotyczące specyficznej części systemu od określonego użytkownika.

Najczęściej wykorzystywane w procesie pozyskiwania wymagań narzędzia to:

Zadawanie pytań – Identyfikacja odpowiedniej osoby, takiej jak klient czy użytkownik i zadanie pytań o wymagania tworzonego systemu.

Obserwacje i wnioskowanie – Obserwacja zachowania użytkowników istniejącego systemu (automatycznego, czy ręcznego), a następnie wnioskowanie o potrzebach na podstawie poczynionych obserwacji.

Dyskusje i opracowania – Dyskusje z użytkownikami na temat ich potrzeb i wspólne formułowanie wymagań.

Negocjacje z uwzględnieniem zbioru standardowego – Rozpoczynanie ustaleń od standardowym zbioru wymagań i własności systemu. Prowadzenie negocjacji z użytkownikami, które z tych standardowych wymagań powinny być włączone, odrzucone bądź zmodyfikowane.

Badanie i identyfikowanie problemów – Przeprowadzenie dochodzenia mającego na celu wykrycie problemów generujących wymagania udoskonalające system. Na przykład jeżeli system jest zbyt wolny, może wymagać złożonego procesu monitorowania wydajności mającego na celu zidentyfikowanie zmian w systemie.

Odkrywanie poprzez twórczy proces – w przypadku bardzo złożonych problemów, nieposiadających oczywistych rozwiązań można zastosować procesy twórcze, w których uczestniczą projektanci i użytkownicy.

Postulowanie – w przypadku braku dostępu do użytkownika lub klienta lub przy tworzeniu bezprecedensowego produktu, można wykorzystać twórczy proces, intuicję projektantów, na podstawie których określa się właściwości i możliwości projektowanego systemu.

W celu zilustrowania technik pozyskiwania wymagań rozważmy projekty, z jakimi spotkali się nasi studenci Pat, Terry i Chris.

Pat zetknął się ze stosunkowo częstym zadaniem pozyskania wymagań. Najlepszą techniką jest najprawdopodobniej dyskusja i zdefiniowanie wymagań wspólnie z użytkownikami. Technika o nazwie Joint Application Design (JAD) jest jedną z tych, które pasują do tego typu sytuacji. Jest to technika szeroko stosowana przy tworzeniu systemów informacyjnych. Technika, która w tym przypadku nie znalazłaby zastosowania, byłoby postulowanie.

Terry niewątpliwie będzie musiał przedyskutować wymagania z inżynierami projektującymi podzespoły odpowiedzialne za kontrolę lotu samolotu. Przydana może być również obserwacja pilotów podczas lądowania. Ze względu na to, że tego typu oprogramowanie nie ma sobie podobnych, techniki takie jak negocjacje z uwzględnieniem zbioru standardowego czy obserwacje i wnioskowanie na podstawie istniejącego systemu są wykluczone.

Chris ma prawdopodobnie najtrudniejszy orzech do zgryzienia, mimo tego że końcowe wymagania będą prawdopodobnie mniej skomplikowane niż w przypadku aplikacji Terry'ego. Technika postulowania może być niezbędna, jeżeli zostanie podjęta decyzja o stworzeniu oprogramowania, które nie ma istniejącego odpowiednika. Jeżeli produkt będzie miał za zadanie konkurować z istniejącymi na rynku pakietami, przydatna może być technika obserwacji i wnioskowania na podstawie istniejącego systemu. Najmniej użyteczną techniką w tym wypadku jest technika zadawania pytań użytkownikom, którzy przecież nie zostali jeszcze zidentyfikowani.

Należy zauważyć, że w rzeczywistych projektach nie ma najlepszej techniki. Inżynier oprogramowania musi umieć wybrać taki zbiór technik, który najbardziej pasuje do konkretnego przypadku.

W dalszym ciągu rozdziału przyjrzymy się kilku technikom pozyskiwania wymagań z podkreśleniem problemów, dla rozwiązania, których powstały.

5.1 Techniki wysokopoziomowe

Techniki wysokopoziomowe określają ogólne ramy dla procesu wydobywania wymagań.

5.1.1 Joint Application Design (JAD).

Technika ta promuje współpracę, zrozumienie i pracę grupową, w którą zaangażowani są klienci, użytkownicy oraz twórcy systemu. Założeniem techniki jest ułatwienie procesu tworzenia wspólnej wizji systemu. Przy jej wykorzystaniu twórcy systemu pomagają użytkownikom w sformułowaniu swoich problemów oraz wypracowaniu rozwiązań. Dzięki temu użytkownicy mają poczucie zaangażowania w proces wytwarzania systemu oraz zyskują wiarę w sukces przedsięwzięcia.

Technika JAD kieruje się czterema podstawowymi zasadami: dynamika grupowa (wykorzystanie pomocniczych spotkań grupowych w celu uwydatnienia indywidualnych potrzeb); wykorzystanie technik wizualnych w celu poprawy komunikacji i wzajemnego porozumienia); stworzenie zorganizowanego i racjonalnego procesu pozyskiwania wymagań oraz zarządzanie tym procesem; oraz wykorzystywanie technik dokumentowania opartych na filozofii WYSIWYG (what you see is what you get) – tak jak to widzisz takie będzie w rzeczywistości (standardowe formularze wypełniane i podpisywane przez wszystkich uczestników spotkań).

Technika JAD ma dwa podstawowe kroki zwane JAD/Plan oraz JAD/Projekt (ang. JAD/Design). Pierwszy krok obejmuje proces pozyskiwania wymagań, drugi proces projektowania oprogramowania.

Każdy krok JAD składa się z trzech faz: *Dostosowanie* (ang. *customization*), *sesja* (ang. *session*) oraz *konsolidacja* (ang. *wrap-up*). Faza dostosowania składa się z zadań przygotowawczych – budowanie zespołu, dopasowywanie procesu do konkretnego systemu, przygotowywanie materiałów. Faza sesji składa się z kilku pomocniczych spotkań twórców i użytkowników. To właśnie podczas tych spotkań rozpoznaje się i dokumentuje wymagania (oczywiście dla kroku JAD/Plan). W fazie konsolidacji zebrane wcześniej informacje przekształca się w formalny dokument – specyfikację wymagań.

5.1.2 Adaptive Loop Framework (ALF)

Technika ALF przypomina w swoich założeniach JAD. Tutaj również mamy do czynienia z ogólnymi ramami procesu, którego zadaniem jest stworzenie powiązań pomiędzy użytkownikami, twórcami oraz procesem wytwarzania systemu informatycznego. Nazwa techniki (Adaptive Loop Framework – cykliczny proces adaptacyjny) pochodzi od idei, która mówi, że proces pozyskiwania wymagań powinien być cykliczny i w każdym przebiegu użytkownicy mają szansę odkrycia nowych, wcześniej niezauważonych wymagań.

Metoda przewiduje trzy tzw. cykle uczące. Twórcy w asyście użytkowników tworzą nowy punkt widzenia na system i wymagania, które go dotyczą. Użytkownicy ‘uczą się’, poprzez to system.

Technika ALF jest szczególnie użyteczna w sytuacjach, gdy istnieją problemy z artykulacją problemów, jak również w przewyciężaniu technicznych problemów występujących przy projektowaniu złożonych systemów.

5.1.3 Prototypowanie

W pewnych sytuacjach, użytkownik jest w stanie lepiej wyrazić swoje potrzeby względem systemu, gdy ma możliwość porównania go z istniejącym lub referencyjnym rozwiązaniem. Kiedy taki system nie istnieje, można wykorzystać technikę Prototypowania. Tworzony jest wtedy prototyp systemu, który pokazuje istotne (z punktu widzenia twórców) cechy systemu. W trakcie interakcji z prototypem użytkownik może poznać swoje prawdziwe wymagania związane z końcowym produktem.

Proces prototypowania rozpoczyna się od wstępnego zapoznania się z potrzebami użytkowników. Następnie następuje iteracyjny proces tworzenia prototypu systemu, którego kolejne wersje przedstawiane są użytkownikowi. Każda iteracja umożliwia użytkownikowi rozpoznanie kolejnych wymagań oraz weryfikacja tych, które w zaimplementowane zostały w prototypie. Ostatecznie, tworzony jest końcowy zbiór wymagań a prototyp zostaje porzucony. Należy zwrócić uwagę na to, iż technika prototypowania ma tylko wtedy sens, gdy dostępne są narzędzia, pozwalające na stworzenie prototypu systemu znacznie szybciej niż miałyby to miejsce w przypadku produktu końcowego. Istnieją narzędzia, które wspierają właśnie takie „szybkie prototypowanie” (RAD – Rapid Application Development).

Technika ta niesie ze sobą również pewne niebezpieczeństwa. Prototyp, który widzi użytkownik jest porzucany i cały proces tworzenia jest zazwyczaj rozpoczynany od początku, po określeniu wymagań. Użytkownicy mogą być zaskoczeni, że widzieli już działający system a muszą jeszcze tyle czekać na końcową wersję.

5.2 Techniki szczegółowe

Techniki szczegółowe zajmują się najczęściej specyficznymi aspektami procesu pozyskiwania wymagań stanowiąc swego rodzaju przewodnik użytkownika.

5.2.1 Burza mózgów (*ang. Brainstorming*)

Jest to prosta technika przeznaczona dla grupy osób, której celem jest wygenerowanie pomysłów. Pozwala uczestnikom na wypowiedzanie sugestii, odkrywanie nowych idei w atmosferze wolnej od krytycyzmu i osądzania.

Burza mózgów powinna się odbywać w grupie 4 – 10 osób. Jedna z nich pełni rolę lidera, którego zadaniem jest rozpoczęcie sesji.

Sesja składa się z dwóch części. Fazy *generacji* i fazy *konsolidacji*. W pierwszej fazie uczestnicy mają za zadanie wykreowanie jak największej liczby pomysłów, bez merytorycznej dyskusji nad ich sensem. Faza druga (konsolidacji) ma na celu przejrzenie, przedyskutowanie oraz odpowiednie pogrupowanie koncepcji.

Taka technika daje możliwość stworzenia wielu punktów widzenia na system. Jest szczególnie przydatna we wczesnej fazie procesu pozyskiwania wymagań.

Technika pozwala również na pokonanie barier poznawczych uczestników, brak krytycyzmu może również dobrze wpływać na problemy związane z barierami komunikacyjnymi.

5.2.2 Wywiady

Jest to ważna technika pomagająca pozyskać szczegółowe informacje od pojedynczych osób. Szeroko stosowana w procesie wydobywania wymagań dla złożonych systemów, często traktowana jako część innych – bardziej ogólnych technik. Może być również przydatna przy małych projektach jako jedyna metoda pozyskiwania wymagań.

Technika wywiadów to nie tylko zadawanie pytań. Jest to bardziej złożona technika, której można nauczyć, a inżynierowie oprogramowania mogą, treningowi i praktyce, czerpać z niej wiele korzyści. Wymaga ona poznania wielu aspektów socjologicznych, umiejętnego słuchania i taktyk prowadzenia wywiadów.

Odpowiednio wyszkolona osoba jest w stanie przy pomocy tej techniki umożliwić użytkownikowi poznanie swoich wymagań, pokonując bariery komunikacyjne.

5.2.3 PIECES

Bardzo często, w przypadku niedoświadczonego analityka, podstawowym problemem jest odpowiedź na pytanie „jak zacząć”. Nie jest również oczywiste, jakie pytania powinny być zadawane. Technika PIECES pomaga rozwiązać powyższe problemy dostarczając zbioru kategorii, które pomagają analitykowi w usystematyzowaniu procesu pozyskiwania wymagań.

Nazwa PIECES jest akronimem angielskich słów reprezentujących sześć kategorii: Wydajność (**P**erformance), informacja i dane (**I**nformation and **d**ata), ekonomiczność (**E**conomy), kontrola (**C**ontrol), sprawność (**E**fficiency), usługi (**S**ervices). Każda z kategorii jest podzielona na podstawowe kwestie, które powinny być wzięte pod uwagę. Technikę można dostosować do konkretnego typu systemu, który ma być zaprojektowany.

Technika ta najlepiej sprawdza się podczas analizowania istniejącego systemu (automatycznego bądź ręcznego), szczególnie, jeżeli proces pozyskiwania wymagań ma na celu ulepszenie istniejącego systemu informatycznego. Tak jak w przypadku techniki

wywiadów pomaga przezwyciężyć problemy komunikacyjne pomiędzy użytkownikami i twórcami.

6. Podsumowanie

Etap pozyskiwania wymagań jest niewątpliwie krytycznym dla całego procesu konstrukcji oprogramowania. Proces ten musi być czymś więcej niż tylko zbieraniem informacji. Powinien stanowić wspólny wysiłek, którego efektem jest umożliwienie wszystkim uczestnikom lepszego poznania systemu, a co za tym idzie podjęcia lepszych decyzji projektowych.

Proces pozyskiwania wymagań napotyka wiele trudności. Uświadomienie sobie ich istnienia jest pierwszym krokiem do jego ulepszenia.

W inżynierii oprogramowania wykorzystuje się wiele technik pozyskiwania wymagań. Dobry specjalista zna co najmniej kilka z nich, potrafi wskazać ich zalety i wady. Dzięki temu jest w stanie dobrać najodpowiedniejszą technikę do konkretnej sytuacji.