Baza wiedzy dla firm programistycznych ze wspomaganiem analizy ryzyka

Mariusz Trzaska (mtrzaska@polbox.com)
PJWSTK

Alina Stasiecka (alas@ipipan.waw.pl)
IPI PAN

Ewa Stemposz (ewag@ipipan.waw.pl.)
PJWSTK, IPI PAN

Kazimierz Subieta (subieta@ipipan.waw.pl.)
PJWSTK, IPI PAN

Streszczenie

Artykuł poświęcono problematyce zarządzania wiedzą ludzką w procesie konstrukcji SI. Wiedza jest ogółem wiadomości niezbędnych do efektywnej i skutecznej realizacji tego rodzaju przedsięwzięć, które, ze względu na swoją specyfikę, są obarczone dużym prawdopodobieństwem niepowodzenia. Wiedza dzieli się na wiedzę jawną oraz wiedzę cichą. Przedstawiono system wspomagający zarządzanie wiedzą cichą, z uwzględnieniem analizy ryzyka wynikającego z niektórych aspektów tej wiedzy. System powstał na bazie systemu relacyjnego w technologii internetowej jako praca dyplomowa studentów PJWSTK.

1. Wprowadzenie

Zarządzanie wiedzą (*knowledge management*) jest relatywnie nową dziedziną informatyczną, która w ostatnich latach znalazła spore nagłośnienie w literaturze komercyjnej. W pewnym stopniu przedmiot tej dziedziny - czyli wiedza - wydaje się tak rozległy i wieloaspektowy, że wyodrębnianie specjalnego *zarządzania* dla tej wiedzy może być uważane za próbę nazywania rzeczy oczywistych. W istocie, wszystkie procesy informacyjne dziejące się w danej organizacji - przechowywanie i przetwarzanie danych, tworzenie, obieg i przetwarzanie dokumentów, procesy administracyjne, itd. - można nazwać zarządzanie wiedzą. Z drugiej jednak

strony, pojawienie się tej dziedziny jest konsekwencją nowej świadomości wśród specjalistów z zakresu organizacji i zarządzania. Poprzez lata starano się sformalizować procesy informacyjne zachodzące w przedsiębiorstwach i organizacjach. Z tą formalizacją wiązano bezpośrednio sukcesy biznesowe, czego skutkiem było skupienie się na technologiach informatycznych, wręcz fetyszyzacja tych technologii. Jak się okazało, nie da się jednak udowodnić bezpośredniego związku pomiędzy sukcesem biznesowym firmy, a stopniem formalizacji lub informatyzacji zachodzących w niej procesów. Gdzie jest więc klucz do sukcesu? Uświadomienie tych okoliczności prowadzi do wniosków, że niekoniecznie znajduje się on w tej części wiedzy, która jest jawna, sformalizowana, zapisana w dokumentach i bazach danych, podlegająca formalnym regułom przetwarzania. Sukces biznesowy w równym, a może nawet w większym stopniu zależy od wiedzy niejawnej, nieprzekładalnej na jakikolwiek język lub kod, będącej własnością umysłów ludzi będących uczestnikami procesów biznesowych.

Zarządzanie wiedzą[1] dotyczy *wiedzy jawnej* (*explicit knowledge*) oraz *wiedzy cichej* (*tacit knowledge*) (patrz np. przeglądowy artykuł [5] oraz bibliografię zawartą w tym artykule). Wiedza jawna może być zapisana w postaci symboli, przetwarzana cyfrowo, oraz zakomunikowana innym uczestnikom procesów biznesowych. Wiedza cicha jest zdobywana w praktyce, jest kwintesencją doświadczenia, wyczucia technologii, bliżej nieokreślonych asocjacji, uświadomionych lub nieuświadomionych poglądów, trenowanych umiejętności rozpoznawania spraw i sytuacji, itd. Ten rodzaj wiedzy w swojej zasadniczej części jest nieprzekazywalny innym i nie daje się zapisać na mediach papierowych lub elektronicznych. Wiedza cicha jest konsekwencją lat nauki, zdobywania doświadczeń, oraz bliżej nieokreślonej umiejętności budowania wewnątrz swojego umysłu niewyrażalnych językowo asocjacji oraz reguł wnioskowania i podejmowania decyzji.

Wiedza cicha, w odróżnieniu od wiedzy jawnej, jest wiedzą poszczególnych ludzi, a nie organizacji jako całości. Jest ona jednak kluczowym strategicznym czynnikiem decydującym o konkurencyjności organizacji na rynku. Wiedza cicha, mimo że tkwiąca w umysłach poszczególnych osób, jest wiedzą obiektywną - może być sprawdzana, testowana i badana empirycznie ze względu na jej jakość i skuteczność. Z punktu widzenia organizacji jest więc istotne, aby wprząc tę indywidualną wiedzę pracowników w ogólną kulturę organizacyjną, do której oni należą. Ten proces zachodzi w dwóch wymiarach: zamianie wiedzy cichej na wiedzę jawną, oraz zamiany wiedzy na poziomie personalnym na wiedzę na poziomie organizacji. Obydwa te wymiary podlegają ograniczeniom. Jak mówi Sveiby (patrz [5]), wiedza jawna jest tylko „wierzchołkiem góry lodowej", w związku z tym organizacje powinny koncentrować się na wykorzystaniu wiedzy cichej w sposób zdyscyplinowany. Nie przeczy to poglądom, że organizacja nie powinna uzależniać się od wiedzy poszczególnych jej pracowników, ponieważ zwiększa to ryzyko w przypadku ich odejścia. Obiektywizm cichej wiedzy oraz umiejętne wprzęgnięcie jej w kulturę organizacyjną mogą tworzyć układ, który nie musi być wrażliwy na losowe czynniki związane z poszczególnymi uczestnikami procesów biznesowych.

W [5] podano siedem podstawowych sugestii dla organizacji dotyczących praktycznych implikacji podziału na wiedzę jawną i wiedzę cichą:

1. Większe skupienie się na wiedzy cichej. Niedocenianie, ignorowanie tej wiedzy może mieć negatywne konsekwencje dla konkurencyjności organizacji na rynku.
2. Nacisk na kolektywną bazę wiedzy całej organizacji. Wiedza ta jest sumą wiedzy jawnej i cichej, i oba te rodzaje wiedzy są istotne dla sukcesu biznesowego.
3. Skupienie się na procesach innowacyjnych, które tworzą przewagę organizacji na rynku.

[1] W sztucznej inteligencji używa się terminów *zarządzanie wiedzą* i *baza wiedzy* w innym znaczeniu, które stara się forsować pogląd, że jest to coś więcej niż proste *zarządzanie danymi* i prosta *baza danych*. W istocie sprowadza się to do użycia pewnych metod formalnych, np. logiki matematycznej. To samo można (najczęściej lepiej) zaprogramować np. typowymi środkami relacyjnej bazy danych, zatem można mieć wątpliwości, czy użycie tych metod formalnych istotnie wnosi nową jakość. To inne znaczenie tych terminów nie ma oczywiście związku z tematem tego artykułu.

4. Skupienie się na ciągłych procesach usprawniania funkcjonowania organizacji, które w zasadniczy sposób są uzależnione od cichej wiedzy.
5. Skupienie się na procesach kształcenia, ponieważ wiedza, zarówno jawna jak i cicha, jest bezpośrednią ich konsekwencją.
6. Skupienie się na kompetencji kolektywnej (społecznej), czyli uzupełnianie się wiedzy różnych osób wewnątrz organizacji i jej części.
7. Rozwój myślenia systemowego, które włącza m.in. rozumienie relacji pomiędzy częścią i całością systemu, relacji wewnątrz systemu, wzorców zachowania się wewnątrz systemu oraz związków systemu z jego otoczeniem.

Technologie informatyczne w zasadniczy sposób wspomagają wiedzę jawną oraz wszelkie procesy związane z tą wiedzą. Czy jest możliwe informatyczne wspomaganie dla wiedzy cichej, skoro ta wiedza nie jest zapisywalna na nośniki komputerowe? Zdyscyplinowane zarządzanie taką wiedzą może polegać na utworzeniu bazy meta-wiedzy, ewidencjonującej, przechowującej, kontrolującej i przetwarzającej informacje dotyczące cichej wiedzy poszczególnych pracowników firmy. Z taką bazą meta-wiedzy można byłoby powiązać pewne narzędzia analityczne, które na postawie informacji o danej organizacji, jej otoczeniu, celach i uwarunkowaniach biznesowych dostarczałyby podstaw dla niektórych decyzji.

Temat stworzenia i praktycznego wdrożenia tego rodzaju bazy meta-wiedzy jest całkowicie nowy, rodzący różnorodność zasadniczych pytań i problemów, w szczególności:

- Jaki rodzaj działalności biznesowej wymagałby takiego wspomagania?
- Jakie rodzaje wiedzy pracowników warto ewidencjonować i w jaki sposób?
- W jaki sposób kontrolować i zarządzać taką wiedzę?
- Jakie decyzje biznesowe mogłyby być uzależnione od badania stanu tego rodzaju wiedzy?
- Jakie narzędzia analityczne (dające pewien syntetyczny wynik) można byłoby skonstruować i jakich efektów można z nich oczekiwać?
- Jak spowodować, aby tego rodzaju baza meta-wiedzy stała się efektywnym środkiem stosowanym w codziennej praktyce biznesowej?

Celem projektu dyplomowego w PJWSTK było stworzenie prototypu komputerowego systemu zarządzania cichą wiedzą, który dawałby szansę uzyskanie początkowych odpowiedzi na niektóre z wymienionych pytań. Oczywiście, przestrzeń swobody wyborów i decyzji przy konstruowaniu tego rodzaju systemu jest przeogromna, zaś wzorce w tym zakresie praktycznie nie istnieją. Projekt należało więc ukierunkować w taki sposób, aby jego cele były jasne, zaś funkcje dobrze zdefiniowane. Ta orientacja spowodowała podjęcie pierwszych decyzji, mianowicie, że projekt będzie dotyczyć firmy programistycznej (której specyfikę zespół projektowy znał stosunkowo najlepiej) i będzie wspomagał zarządzanie projektami programistycznymi. Spośród wielu aspektów zarządzania powiązanych z cichą wiedzą wybrano aspekt zarządzania ryzykiem, uznając, że brak pełnej świadomości kierownictwa projektu odnośnie stanu cichej wiedzy pracowników, zła struktura tej wiedzy, niesie za sobą zagrożenia dla projektów.

Wiedzę cichą informatyka można określić jako zasób wiadomości, który dany pracownik uzyskał po ukończeniu studiów, wiedzę o nowościach informatycznych, o rynku pracy, o prowadzonych w danym przedsiębiorstwie projektach, itp. Do tej wiedzy zalicza się także doświadczenia zdobyte w toku prowadzenia pewnej działalności, np. podczas poprzednich projektów. Do tej wiedzy zaliczane są także umiejętności nabyte w czasie pracy, np. zdolność zainstalowania serwera MS SQL V6.5 lub zastosowania narzędzia Rational Rose. Wiedza nabyta, doświadczenia i umiejętności składają się na to, co będziemy dalej uważali za wiedzę użyteczną dla efektywnej i skutecznej realizacji prowadzonych przedsięwzięć i przechowywaną w przedmiotowym systemie. Przechowywanie wiedzy pociąga za sobą problemy związane z jej

porządkowaniem. Większa skala przedsiębiorstwa lub organizacji rodzi nieproporcjonalnie więcej problemów z dostępem do wiedzy, zarówno cichej jak i jawnej, i jej porządkowaniem.

W projekcie założono, że jednym z wymiernych efektów wdrożenia systemu zarządzania wiedzą powinno być zarządzanie ryzykiem. Jest ono definiowane jako zestaw metod i praktyk umożliwiających identyfikację, przeprowadzenie analizy, ustalenie hierarchii oraz wypracowanie reguł prewencyjno-naprawczych dla każdego rodzaju zagrożenia projektu. Przedsięwzięcia informatyczne (projekty), ze względu na swoją specyficzną naturę: ogromną złożoność, interdyscyplinarność, niski stopień powtarzalności, niedojrzałość technologii, itp. należą do przedsięwzięć obdarzonych szczególnie dużym ryzykiem niepowodzenia. Zarządzanie ryzykiem, podejmowanie działań prewencyjnych lub mitygujących ryzyko, oraz działań naprawczych po wystąpieniu ryzyka jest przedmiotem norm jakości oraz wielu metodyk i zaleceń (patrz np. [7]). Dotyczy to w szczególności norm z serii ISO 9000, modelu CMM, metodyki Prince-2, normy regulującej cykl życia oprogramowania ISO 12207, norm IEEE-730, klasyfikacje i zalecenia Software Engineering Institute, itd.

Przykładem narzędzia do analizy ryzyka może być SRE opisany w [9]. Tego rodzaju metod, narzędzi, zaleceń i innych materiałów jest sporo, patrz np. [1,2,3,4,6,8]. Warto tu zwrócić uwagę na pewną okoliczność. Metody zarządzania ryzykiem opierają się na predefiniowanych listach zagrożeń, subiektywnej konkretyzacji takiej listy dla danego przedsięwzięcia, subiektywnych ocenach prawdopodobieństw zagrożeń i ocenach ich skutków. Tym ocenom podporządkowane są subiektywnie wybrane działania prewencyjne (mitygujące) oraz naprawcze. W powyższych zdaniach istotne jest pojawienie się wielokrotnie określenia „subiektywne", poprzez które staramy się uwypuklić fakt, że niezależnie od wagi, którą zarząd projektu powinien przykładać do analizy ryzyka, jest ono uwarunkowane nieprzywidywalnymi aspektami przyszłości, a przez to musi być obarczone błędem wynikającym z subiektywnych ocen odnośnie stanu obecnego oraz stanu przyszłego.

Jest istotne, aby przesunąć część tego subiektywizmu na grunt bardziej obiektywnych ocen i decyzji. Baza opisująca cichą wiedzę pracowników stwarza szansę objęcia tego rodzaju zabiegiem pewnej części zagrożeń projektów programistycznych. Do takich zagrożeń należą m.in.:

- Ryzyko uzależnienia firmy od niewielkiej liczby kompetentnych pracowników;
- Ryzyko niedopasowania wiedzy niezbędnej do realizacji projektu do obecnej wiedzy posiadanej przez pracowników;
- Ryzyko nieprzystosowania kadrowego firmy do obecnych tendencji technologicznych;

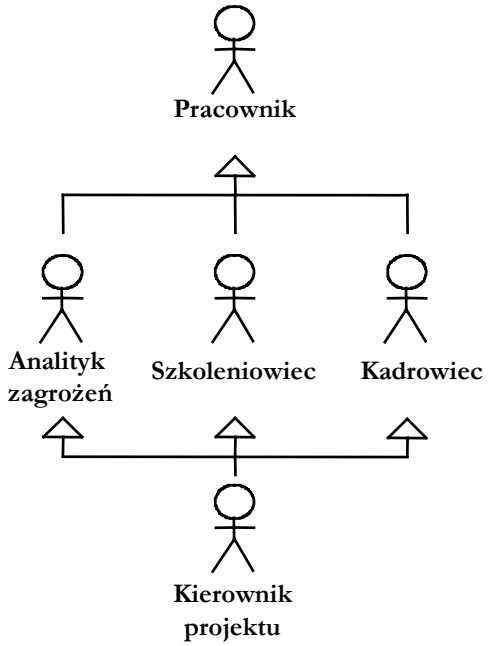
System został zrealizowany przez studentów PJWSTK w technologii klient - serwer opartej na relacyjnej bazie danych i technologiach Internetowych. Po stronie serwera znajdują się dwie warstwy: w*arstwa biznesowa* oraz w*arstwa bazy danych.* Warstwa biznesowa jest łącznikiem pomiędzy warstwą klienta i warstwą bazy danych. W jej skład wchodzi serwer WWW z ASP oraz serwer FTP. Warstwa bazy danych jest obsługiwana przez MS SQL Server 7, która odpowiada na zapytania warstwy biznesowej nie kontaktując się bezpośrednio z warstwą klienta. Dostęp do bazy danych jest realizowany przez sterownik ODBC, będący elementem technologii ADO. Powyższe warstwy są zrealizowane na tym samym komputerze. Warstwa klienta wykonuje dwa zadania: wysyłanie zapytania do warstwy biznesowej oraz przedstawienie wyników. Klient może wykorzystać dowolną przeglądarkę internetową, która otrzymuje od warstwy biznesowej gotowe strony hipertekstowe oraz daje możliwość wysyłania zapytań. Rozwiązanie daje możliwość wykorzystania przez klienta dowolnej platformy sprzętowej.

W dalszej części artykuł jest skonstruowany następująco. W Rozdziale 2 przedstawimy nasze działania zmierzająca do rozpoznania potencjalnych potrzeb użytkowników systemu. Rozdział 3 opisuje model pojęciowy informacji pamiętanych w systemie. Rozdział 4 poświecony jest opisowi modułu analizy zagrożeń. Rozdział 5 zawiera podsumowanie i plany związane z tym projektem.

2. Rozpoznanie potrzeb użytkowników systemu

W celu rozpoznania rynku przeprowadzono szereg wywiadów z pracownikami różnych firm i organizacji. Przygotowano ankietę, zawierającą kilkadziesiąt pytań. Po przeanalizowaniu ankiet postanowiono skupić się na przedsiębiorstwach związanych z produkcją oprogramowania. Potencjalnym użytkownikiem systemu mógłby być pracownik firmy informatycznej, (jak na Rys.1), np. *kierownik projektu* który mógłby szybko i w łatwy sposób uzyskiwać informacje o doświadczeniu analityków, projektantów, programistów i innych osób, które zamierza zaangażować do projektu. Może on także uzyskiwać informacje o podobnych projektach wcześniej prowadzonych, kosztach ich wykonawstwa, zastosowanych rozwiązaniach technologicznych, itp. Wiedza ta pozwoliłaby mu na utworzenie odpowiedniego zespołu specjalistów, wspomogłaby go przy podejmowaniu decyzji oraz uchroniłaby projekt przed zagrożeniami lub zminimalizowała negatywne skutki wystąpienia zagrożeń.

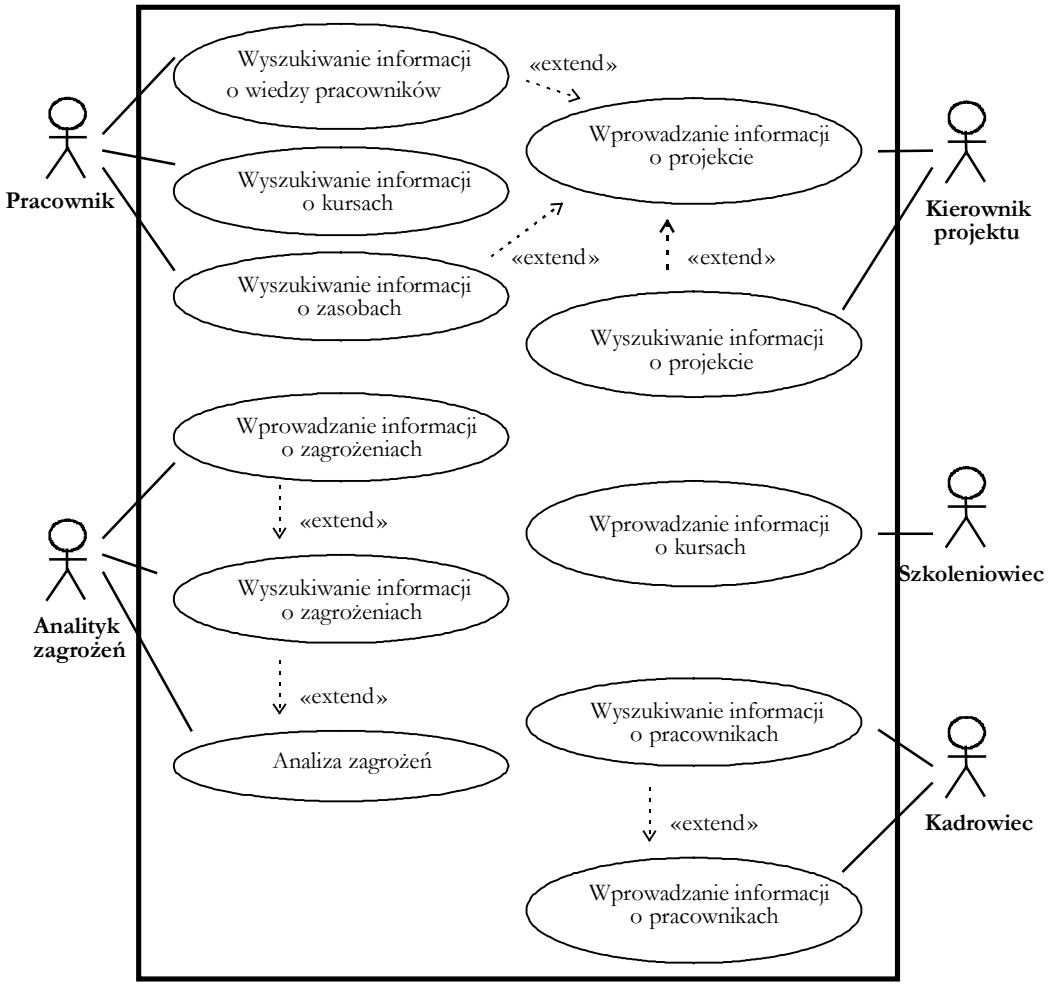
Zakres wiedzy przechowywanej w systemie jest przedmiotem decyzji, które mogą być podjęte na dwóch poziomach. Na pierwszym poziomie projektanci systemu podjęli decyzję odnośnie kategorii cichej wiedzy, której opis będzie znajdować się w systemie. Na drugim poziomie baza wiedzy może być zapełniana z pewnym stopniem swobody przez użytkowników systemu. Tego rodzaju rozwiązanie jest skutkiem kompromisu pomiędzy generycznością systemu (umożliwiającą zapis dowolnej wiedzy) a przystosowaniem jego struktury i interfejsów do specyfiki konkretnych zastosowań. Bardziej generyczny system dawałby większe możliwości w zakresie rejestracji wiedzy, ale też zmniejszałby skuteczność środków przetwarzania tej wiedzy dla konkretnych celów lub zmuszałby użytkownika do kłopotliwych zabiegów przystosowania go do tych celów. System wyspecjalizowany rodzi natomiast trudności z zapisem i przetwarzaniem pewnych rodzajów wiedzy. Stworzony prototyp ustala ten kompromis w pewnym punkcie wybranym na podstawie badań ankietowych, ale oczywiście w wyniku dalszych badań i doświadczeń może ten punkt być przesunięty zarówno w stronę większej generyczności, jak i w stronę większego wyspecjalizowania.



Rysunek 1: Hierarchia dziedziczenia dla aktorów (użytkowników) systemu

Każdy użytkownik systemu, zgodnie z posiadanymi uprawnieniami (patrz Rys.2) mógłby uzyskiwać informacje z zakresu:

- wiedzy: wiedzy ogólnej, doświadczenia oraz umiejętności innych pracowników, w tym informacji o kursach i szkoleniach ukończonych przez pracowników,
- projektów realizowanych i zrealizowanych w firmie,
- aktualnej oferty szkoleń organizowanych w firmie,
- możliwych wystąpień sytuacji awaryjnych,
- zapobiegania sytuacjom awaryjnym,
- postępowania w sytuacjach awaryjnych.



Rysunek 2: Diagram przypadków użycia

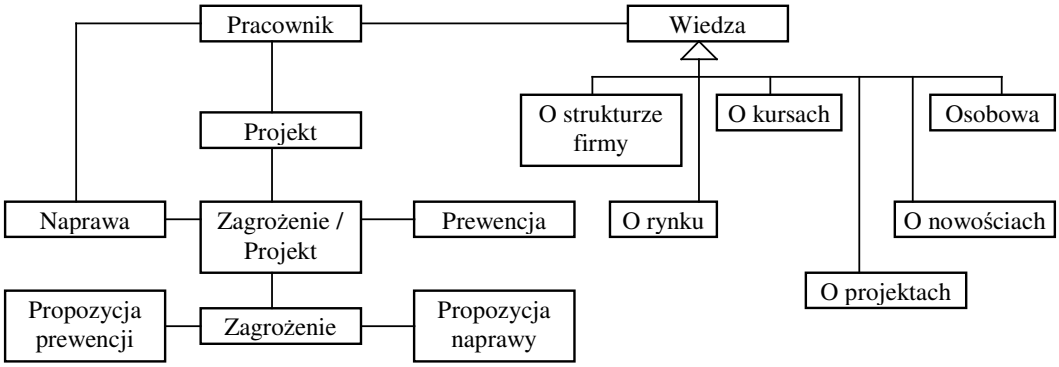
3. Model pojęciowy

Do stworzenia modelu pojęciowego systemu wykorzystane zostało podejście obiektowe z użyciem notacji UML. Poniżej podano opisy najważniejszych zaimplementowanych klas:

Wiedza	Wystąpienia tej klasy, oraz jej klas wyspecjalizowanych, zawierają informacje o wszystkich rodzajach wiedzy przechowywanej w systemie.
Pracownik	Klasa specyfikuje dane personalne osób zatrudnionych w firmie.
Projekt	Klasa przechowuje informacje zarówno o tworzonych, jak również o już zrealizowanych w firmie projektach.
Zagrożenie	Wystąpienia tej klasy specyfikują zagrożenia, które mogą wystąpić w obszarze działalności firmy.
Propozycja prewencji	Klasa służy do opisu czynności, których realizacja może zmniejszyć prawdopodobieństwo wystąpienia danego zagrożenia.
Propozycja naprawy	Klasa specyfikuje zadania naprawcze dla sytuacji, gdy dane zagrożenie wystąpiło.
Zagrożenie/ projekt	Klasa ustalająca listę zagrożeń, których wystąpienie jest dla danego projektu możliwe.
Naprawa	Klasa zawiera opis czynności, które zostały wykonane jako naprawa, po wystąpieniu konkretnego zagrożenia w konkretnym projekcie.

Ogólną ideę przyświecającą twórcom systemu można ująć w kilku punktach:

- **pracownicy** mogą posiadać różne rodzaje **wiedzy**,
- w **projektach** uczestniczą różni **pracownicy** firmy,
- z każdym **projektem** są związane różne **zagrożenia**,
- dla konkretnego **zagrożenia**, przewidziano różne rodzaje **prewencji** oraz **napraw**,
- w przypadku wystąpienia **zagrożenia** w konkretnym **projekcie**, konkretny **pracownik** dokonuje **naprawy**.



Rysunek 3: Fragment uproszczonego diagramu klas związanego z przechowywaniem cichej wiedzy i przeciwdziałaniu zagrożeniom

Do przechowywania informacji o konkretnym zagrożeniu, dla konkretnego projektu, wykorzystywane są obiekty klasy **Zagrożenie/Projekt**. Ich opis znajduje się w tabeli nr 1.

Atrybut	**Znaczenie**
Prawdopo-dobieństwo	Określa prawdopodobieństwo wystąpienia zagrożenia w projekcie
Dkosztów	Określa zmianę kosztów projektu w przypadku wystąpienia zagrożenia
Dczasu	Określa zmianę czasu realizacji konkretnego projektu w przypadku wystąpienia zagrożenia
Djakości	Określa stopień utraty jakości produktu, np. zmniejszenie poprawności, użyteczności, niezawodności, wydajności, itp.
Priorytet	Określa znaczenie, jakie zagrożenie stanowi dla projektu, co pośrednio wynika z trzech poprzednich atrybutów

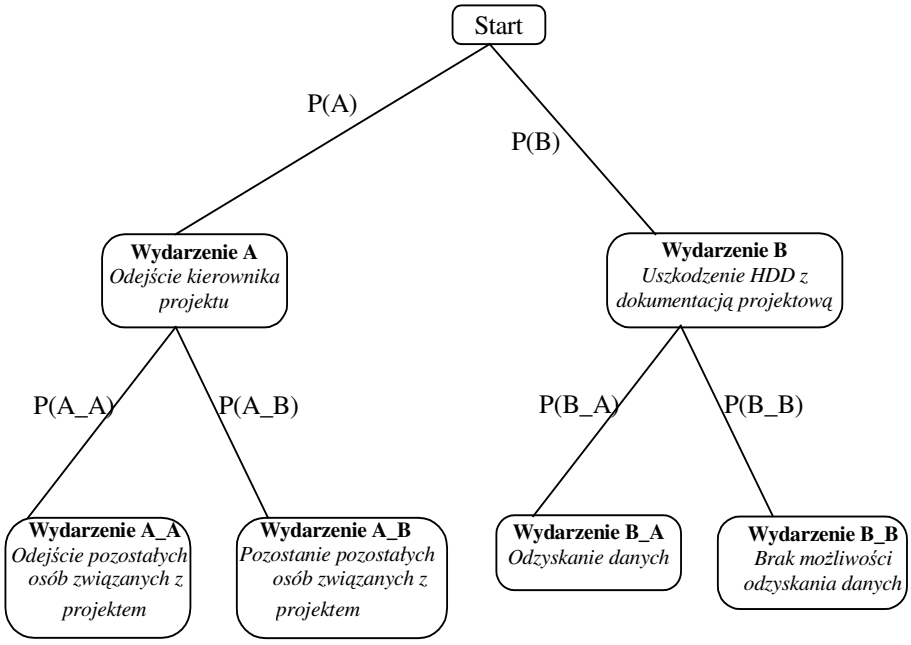
Tabela 1: Opis klasy Zagrożenie / Projekt

4. Analiza zagrożeń

Skuteczne zarządzanie ryzykiem wymaga, aby był to proces ciągły. Nie wystarczy akcja jednorazowa polegająca na zidentyfikowaniu potencjalnych zagrożeń w momencie rozpoczynania realizacji projektu. Należy na bieżąco poddawać ocenie zagrożenia aktualnie istniejące i zagrożenia, które mogą wystąpić w przyszłości jako efekt już istniejących czy też zupełnie od nich niezależnie, a to z kolei wymaga posiadania odpowiedniej infrastruktury. W opisywanym systemie taką infrastrukturę tworzą predefiniowane raporty, które nie tylko prezentują wiedzę zawartą w systemie, z założenia w czytelnej, klarownej dla użytkownika postaci, ale umożliwiają także przeprowadzenie symulacji różnych sytuacji, które mogą wystąpić w trakcie realizacji przedsięwzięcia programistycznego. Raporty stanowią istotne wspomaganie dla analityka zagrożeń i kierownika projektu w podejmowaniu najbardziej odpowiednich dla danej sytuacji

decyzji. Przykładowe raporty wbudowane w system są w szczególności następujące: analiza stopnia wykorzystania budżetu, symulacja wykorzystania budżetu projektu, analiza statusu pracownika, analiza aktualnego stanu zasobów ludzkich.

W dalszej przyszłości planuje się rozbudowę systemu o drzewa ryzyka, które pozwolą na dokładniejszy opis wydarzeń poprzez uwidocznienie ich wzajemnych zależności. Na bazie wystąpienia konkretnego zagrożenia można będzie lepiej określić jego wpływ na dalszy przebieg projektu i wcześniej podjąć przewidziane w systemie czynności prewencyjne, np. w przypadku zagrożenia odejściem kierownika projektu, można przez podniesienie wynagrodzeń pozostałym członkom zespołu, zmniejszyć negatywne skutki tego wydarzenia.



Rysunek 4: Przykładowe drzewo ryzyka

Trudność stanowi tu sposób szacowania poszczególnych prawdopodobieństw oraz ocena wpływu wydarzeń na dalsze losy projektu.

5. Podsumowanie i plany

Opisany powyżej w ogromnym skrócie prototyp systemu łączy aktualne tendencje badawcze w zakresie zarządzania cichą wiedzą ludzką w procesie realizacji przedsięwzięć programistycznych z elementami inżynierii oprogramowania oraz z działalnością implementacyjną zmierzającą do stworzenia narzędzia użytecznego w codziennej pracy zespołów projektowych. Orientacja tego narzędzia na zarządzanie ryzykiem może przynieść wymierne korzyści wynikające z lepszej przewidywalności potencjalnych zagrożeń, co jest szczególnie ważne przy projektach angażujących duże zespoły, duże środki finansowe i znaczną infrastrukturę informatyczną. Połączenie elementów ewidencjonowania i zarządzania cichą wiedzą, inżynierii oprogramowania oraz zarządzania ryzykiem jest pewnym arbitralnym wyborem wynikającym z naszego wyczucia tego, gdzie należy oczekiwać największych korzyści z takiego systemu z punktu widzenia obecnej praktyki firm programistycznych. Poruszając się w ogromnej przestrzeni dość dowolnych

wyborów w zakresie celów i własności tworzonego systemu - przy braku wzorców - decyzje projektowe, aczkolwiek świadome, siłą rzeczy nie muszą być w 100% trafne. Zbudowanie tego systemu uświadomiło jego twórcom nowe możliwości rejestracji i wykorzystania informacji o cichej wiedzy pracowników firmy programistycznych. To powoduje chęć rozbudowy i zmiany orientacji tego narzędzia, w szczególności w następujących kierunkach:

- Zwiększenie stopnia generyczności systemu poprzez uwzględnienie dalszych rodzajów rejestrowanej cichej wiedzy oraz jej klasyfikacji, strukturalizacji i weryfikacji;
- Rozbudowa narzędzi analitycznych pozwalających na dostarczenie kierownictwu przedsięwzięć bezpośrednich danych decyzyjnych wynikających ze stanu cichej wiedzy;
- Rozbudowa modułu zarządzania ryzykiem o nowe techniki, które są dyskutowane w literaturze;
- Powiązanie systemu z innymi aspektami procesów biznesowych zachodzących w firmie programistycznej, w szczególności, z rejestracją i monitorowaniem zużycia zasobów (czasu, budżetu, infrastruktury, itd.) podczas realizacji projektów, co mogłoby wspomóc m.in. bieżącą ocenę ryzyk związanych z projektem.

Tematom tym będą poświecone dalsze prace.

Literatura

[1]. B. W. Boehm. Software Risk Management: Principles and Practices. IEEE Software, 8, 1:32-41, 1991

[2]. B. Boehm. Project Termination Doesn't Equal Project Failure, IEEE Computer Vol. 33, No. 7, 2000

[3]. R. N. Charette. The Mechanics of Managing IT Risk. Journal of Information Technology, 11:373-378, 1996

[4]. R. Fairley. Risk Management for Software Projects. IEEE Foftware, 11, 3:57-67, 1994

[5]. J.-A. Johannessen, J.Olaisen, B.Olsen. Mismanagement of Tacit Knowledge: Knowledge Management, the Danger of Information Technology, and What to do About It. http://program.forskningsradet.no/skikt/johannessen.php3, 1998

[6]. D. Karolak. Software Engineering Risk Management, IEEE Computer Society Press, 1996

[7]. C. Mazza et al. Software Engineering Guides. Prentice Hall Europe, 1996

[8]. T. Moynihan. How Experienced Project Managers Assess Risk. IEEE Software, 14, 3:35-41, 1997.

[9]. R.C. Williams, G. J. Pandelios, S. G Behrens. Software Risk Evaluation (SRE) Method Description (Version 2.0). SEI, 1999. http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr029-body.pdf