Baza wiedzy dla firm software'owych 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 w procesie konstrukcji SI. Przez wiedzę rozumie się tu ogół wiadomości niezbędnych do efektywnej i skutecznej realizacji tego rodzaju przedsięwzięć, które – ze względu na swoją specyfikę – jakby "z natury rzeczy" są obarczone dużym prawdopodobieństwem niepowodzenia. Przedstawiono system, wspomagający zarządzanie wiedzą z uwzględnieniem analizy ryzyka, wykonany w technologii internetowej przez studentów PJWSTK jako praca dyplomowa.

1. Wprowadzenie

Wiedzę ogólną można określić jako zasób wiadomości z pewnej dziedziny, czy też dotyczących pewnego zagadnienia, np.: wiedza jaką Kowalski uzyskał po ukończeniu danego kierunku studiów, wiedza o nowościach informatycznych, o rynku pracy, o prowadzonych w danym przedsiębiorstwie projektach, itp. Mianem doświadczenia określimy tu ogół informacji zdobytych w toku prowadzenia

pewnej działalności, np. Nowak pracował przy projektach dla firm księgowych, w związku z czym posiada doświadczenie w tej dziedzinie. Z kolei, umiejętność jest to zdolność wykonania czegoś, np. Abacki instalował serwer MS SQL V6.5, więc posiada umiejętności w tej dziedzinie. Wszystkie trzy, wymienione powyżej pojęcia, składają się na to, co będziemy dalej uważali za wiedzę użyteczną dla efektywnej i skutecznej realizacji prowadzonych przedsięwzięć i przechowywaną w omawianym systemie. Przechowywanie wiedzy pociąga za sobą problemy związane z jej porządkowaniem. Jak pokazuje praktyka, im większe przedsiębiorstwo, tym więcej problemów z porządkowaniem wiedzy, zarówno tej posiadanej przez każdego z pracowników z osobna, jak i tej posiadanej przez przedsiębiorstwo jako całość.

Zarządzanie wiedzą jest bezpośrednio powiązane z zarządzaniem ryzykiem, jeśli zdefiniujemy to pojęcie jako zbiór praktyk umożliwiających identyfikację, przeprowadzenie analizy, ustalenie hierarchii oraz wypracowanie reguł prewencyjno-naprawczych dla każdego rodzaju niekorzystnego wydarzenia, zwanego dalej zagrożeniem. Przedsięwzięcia informatyczne (projekty), ze względu na swoją specyficzną naturę: ogromną złożoność, interdyscyplinarność, niski stopień powtarzalności, niedojrzałe technologie, itp. należą do przedsięwzięć obdarzonych dużym ryzykiem, czyli dużym prawdopodobieństwem niepowodzenia. W ciągu ostatnich lat obserwuje się zwiększone zainteresowanie tą tematyką, o czym świadczą zarówno publikacje wymienione w bibliografii, jak i praktyka. Wydaje się, że warto przyjrzeć się zagrożeniom wcześniej, bo coś co wydaje się mało groźne dzisiaj, jutro może się okazać się poważnym problemem.

Analiza ryzyka może opierać się na połączonych strategiach:

- *Top-down risk estimation* czyli oszacowaniu wpływu poszczególnych zagrożeń, rozpoczynając od "góry" czyli od najbardziej ogólnych zagadnień, aż do terminów wykonania poszczególnych części projektu.
- *Bottom-up risk management* polega na uszczegółowieniu wyników poprzedniej fazy, a w szczególności:
    - nadaniu priorytetów poszczególnym zagrożeniom,
    - podjęciu czynności mających zminimalizować zarówno prawdopodobieństwo wystąpienia, jak i ewentualne skutki poszczególnych zagrożeń,
    - kontrolowaniu postępów poszczególnych etapów.

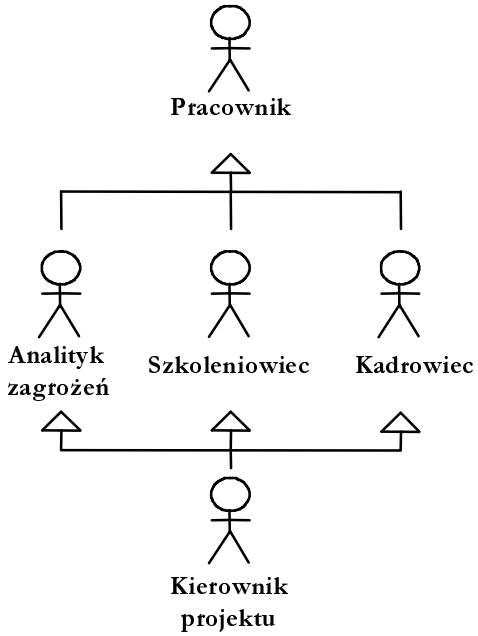
Przykładem narzędzia działającego według powyższych reguł może być TDM schedule metric, opisany w [8].

Wychodząc naprzeciw problemom, które odczuwają duże firmy software'owe, grupa projektowa specjalizacji "*Inżynieria baz danych i oprogramowania"* z PJWSTK, postanowiła skonstruować łatwo dostępny, prosty w obsłudze i bezpieczny dla przechowywanej zawartości, system archiwizacji wiedzy, użytecznej dla efektywnej i skutecznej pracy tego rodzaju firm, połączony z identyfikacją oraz obsługą sytuacji kryzysowych.

W dalszej części artykułu, zdefiniujemy wymagania na system, opiszemy jego architekturę, model pojęciowy oraz moduł analizy zagrożeń.

2. Rozpoznanie potencjalnego użytkownika systemu i jego potrzeb

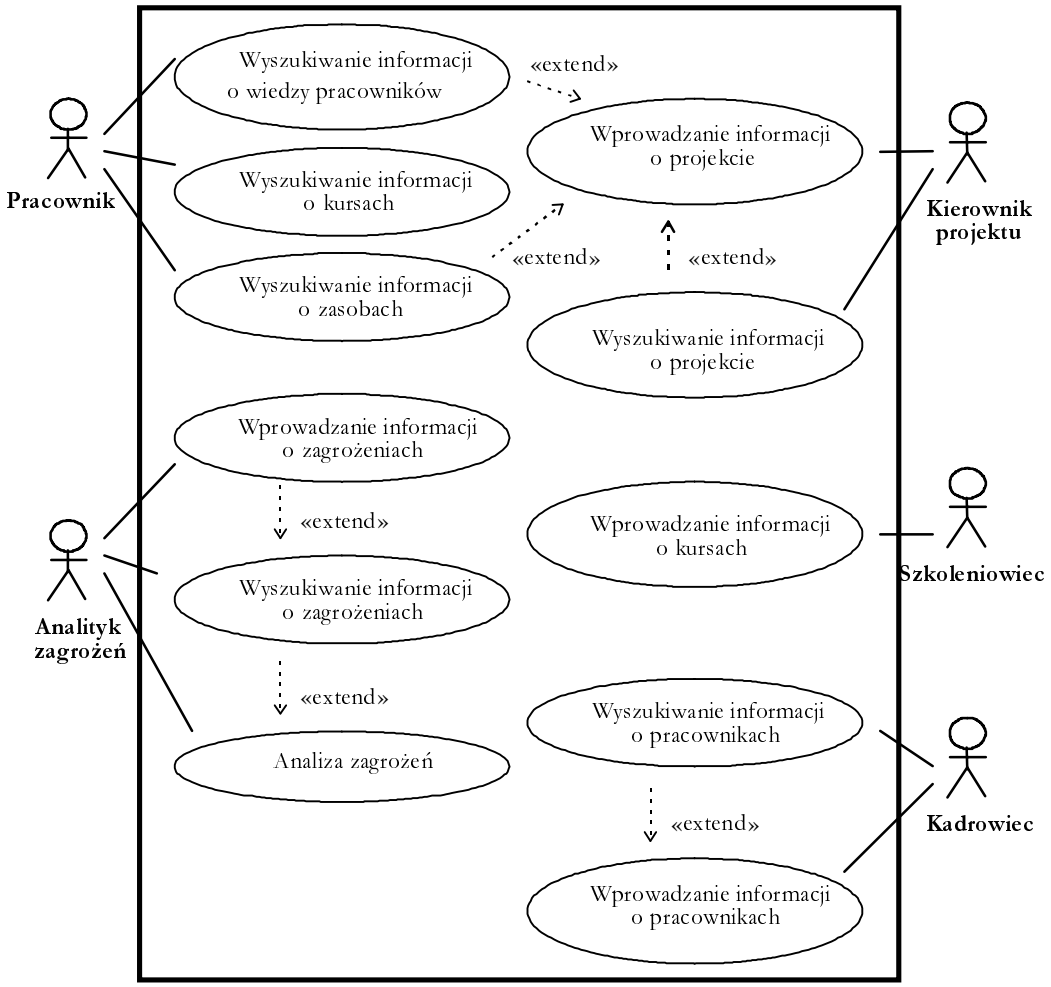
W celu rozpoznania rynku, przeprowadzono szereg wywiadów z pracownikami różnych firm. Przygotowano ankietę, zawierającą kilkadziesiąt pytań. Po przeanalizowaniu ankiet postanowiono skupić się na przedsiębiorstwach związanych z produkcją systemów komputerowych.



Rysunek 1: Hierarchia dziedziczenia dla aktorów (użytkowników) systemu

Potencjalnym użytkownikiem systemu byłby *pracownik* firmy informatycznej, (jak na rys. 1), np. *kierownik projektu* który mógłby szybko i w

łatwy sposób uzyskiwać informacje o doświadczeniu analityków, projektantów, programistów, … których zamierza zaangażować do projektu, o podobnych projektach, kosztach ich wykonawstwa, zastosowanych rozwiązaniach technologicznych, itp. Wiedza ta pozwoliłaby mu na utworzenie zespołu specjalistów, wspomogłaby podejmowanie decyzji oraz uchroniłaby "przed" lub zminimalizowała negatywne skutki "po" wystąpieniu zagrożeń - nieoczekiwanych i niekorzystnych dla realizacji projektu wydarzeń.



Rysunek 2: **Diagram przypadków użycia**

Każdy użytkownik systemu, zgodnie z posiadanymi uprawnieniami (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.

3. Opis architektury systemu

System wykonany przez studentów PJWSTK, działa w technologii klient - serwer. Komunikacja odbywa się poprzez sieć Internet.

3.1 Serwer

Po stronie serwera znajdują się dwie warstwy:

- *Warstwa biznesowa* będąca łącznikiem pomiędzy warstwą klienta i warstwą bazy danych, w jej skład wchodzi serwer WWW z ASP oraz serwer FTP.
- *Warstwa bazy danych,* obsługiwana przez MS SQL Server 7, odpowiada na zapytania warstwy biznesowej, nie kontaktując się bezpośrednio z warstwą klienta. Dostęp do bazy danych jest realizowany przez sterownik ODBC będący jednym z elementów technologii ADO.

W projekcie zastosowano scentralizowany model serwera, tzn. dwie warstwy: biznesowa i bazy danych na serwerze, co przewiduje użycie tylko jednego komputera po stronie serwera.

3.2 Klient

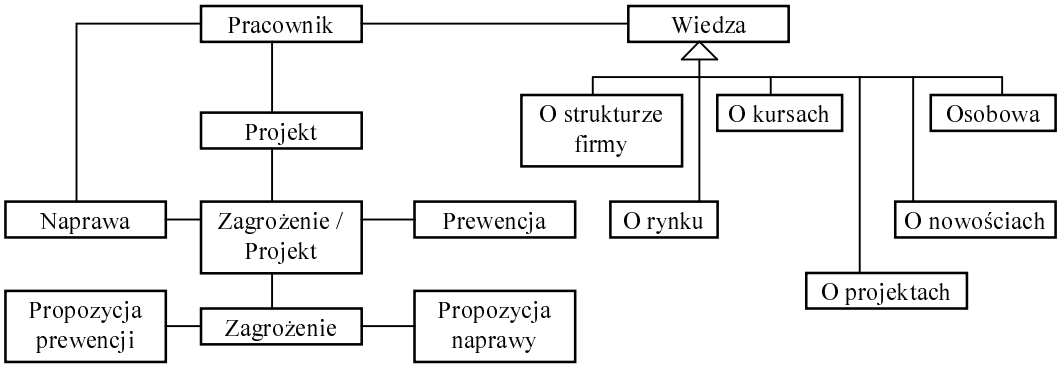
Warstwa klienta ma do wykonania 2 zadania: wysłanie zapytania do warstwy biznesowej oraz przedstawienie wyników zapytań. Do tych celów użyto przeglądarki internetowej, która otrzymuje od warstwy biznesowej gotowe strony hipertekstowe oraz daje możliwość wysyłania do niej zapytań. Takie rozwiązanie pozwala uniezależnić klienta od platformy software'owej.

4. Model pojęciowy

Do stworzenia modelu pojęciowego systemu "Baza wiedzy dla firm software'owych ze wspomaganiem analizy ryzyka" wykorzystane zostało podejście obiektowe, przy użyciu języka UML.

Poniżej podano opisy najważniejszych klas:

Wiedza	Klasa **Wiedza** oraz jej klasy specjalizowane zawierają informacje o wszystkich rodzajach wiedzy przechowywanej w systemie.
Pracownik	Klasa, która specyfikuje dane personalne osób zatrudnionych w firmie.
Projekt	Klasa przechowująca 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 możliwe dla danego projektu.
Naprawa	Klasa zawiera opis czynności, które zostały wykonane jako naprawa, po wystąpieniu konkretnego zagrożenia w konkretnym projekcie.



Rysunek 3: Szkic fragmentu diagramu klas związanego z przechowywaniem wiedzy i przeciwdziałaniu zagrożeniom

Ogólną ideę, przyświecającą twórcom systemu można ująć w kilku punktach:

- **pracownicy** posiadają różne rodzaje **wiedzy**,
- w p**rojektach** 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**.

Do przechowywania informacji o konkretnym zagrożeniu, dla konkretnego projektu, wykorzystywane są obiekty klasy **Zagrożenie/Projekt**. Ich dokładny 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

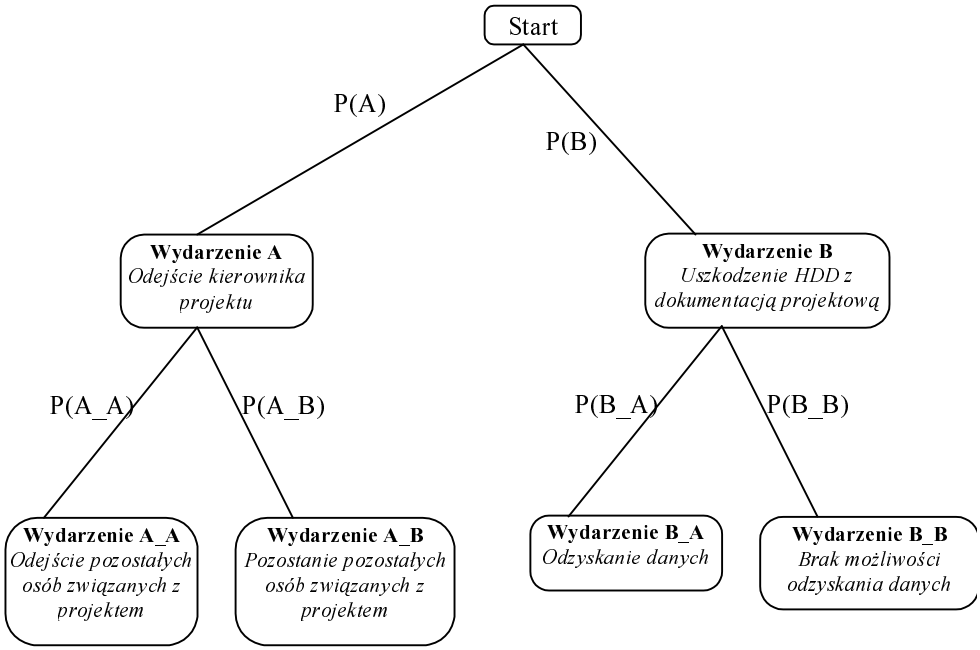
	poprawności, użyteczności, niezawodności, wydajności, itp.
Priorytet	Określa znaczenie, jakie zagrożenie stanowi dla projektu, co pośrednio wynika z trzech poprzednich atrybutów

Tabela 1: **Opis klasy Zagrożenie / Projekt**

5. Analiza zagrożeń

Skuteczne zarządzanie ryzykiem wymaga by 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 i umożliwiają przeprowadzenie symulacji różnych sytuacji, które mogą wystąpić w trakcie realizacji SI. Raporty stanowią istotne wspomaganie dla analityka zagrożeń i kierownika projektu w podejmowaniu najbardziej odpowiednich dla danej sytuacji decyzji. Przykładowe raporty wbudowane w system to: 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.

Wydaje się, że doskonałym sposobem na rozbudowę modułu analizy zagrożeń będzie stworzenie elastycznego generatora raportów. Do jego skonstruowania planowane jest wykorzystanie języka XML. Zakłada się, że XML powinien ułatwić kreowanie indywidualnych sposobów przeprowadzania analiz i prezentowania ich wyników, dzięki możliwościom, które posiada zarówno w zakresie specyfikowania formatu raportu, jak i prezentowania semantyki złożonych danych.

7. Podsumowanie

Wydaje się, że opisany powyżej system jest przykładem dobrego połączenia działalności badawczej z działalnością realizacyjną. Uwzględnianie zarządzania ryzykiem może przynieść wymierne korzyści: projekt będzie trwał krócej, a przez to mniej kosztował, będzie bardziej przewidywalny i będzie lepiej odzwierciedlał potrzeby klientów, bo lepiej można będzie zapanować nad wypełnianiem wyspecyfikowanych wymagań. Jak pokazuje praktyka, zarządzanie ryzykiem przynosi wyraźne korzyści już przy projektach, angażujących co najmniej pięć osób. Wydaje się także, że wytworzony przez studentów PJWSTK produkt może przynieść realne korzyści każdej firmie, niekoniecznie software'owej. Każda

działalność ludzka wiąże się przecież z zagrożeniami, a co za tym idzie z ryzykiem.

8. Bibliografia

[1]. B. W. Boehm. Software risk management: principles and practices. IEEE Software, 8, 1:32-41, 1991

[2]. M. J. Varr, S. L. Konda, I. Monarch, F. C. Ulrich, C. F. Walker. Taxonomy-Based Risk Identification. Technical Report CMU/SIĘ-93-RT-6. Carnegie Mellon University, Software Engineering Institute, Pittsburgh, Pennsylvania, 1993.

[3]. R. Fairley. Risk management for software projects. IEEE Foftware, 11, 3:57-67, 1994

[4]. R. N. Charette. The Mechanics of managing IT risk. Journal of Information Technology, 11:373-378, 1996

[5]. P. Kulik. What is Software Risk Management? KLCI, 1996

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

[7]. M. Myerson. Risk Management Processes for Software Engineering Models. Artech House, 1996

[8]. P. Kulik. Team-Driven Schedule Metrics, KLCI, 1996

[9]. T. Moynihan. How experienced project managers assess risk. IEEE Software, 14, 3:35-41, 1997.

[10]. IEEE Standards Software Engineering Collection, IEEE-SA, 1999

[11]. E. Yourdon. Marsz ku klęsce. Poradnik projektanta systemów. Wydawnictwa Naukowo-Techniczne, Warszawa, 2000.

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

[13]. J. Górski et al. Inżynieria oprogramowania w projekcie informatycznym, Mikom, wydanie II rozszerzone, 2000