

Tomasz GRATKOWSKI

UNIwersytet Zielonogórski Wydział Elektrotechniki, Informatyki i Telekomunikacji, Instytut Informatyki i Elektroniki, ul. prof. Z. Szafrana 2, 65-246 Zielona Góra

Platforma do zarządzania zadaniami wielokrotnego użycia

dr inż. Tomasz GRATKOWSKI

Uzyskał stopień doktora nauk technicznych w zakresie informatyka na Wydziale Informatyki i Zarządzania Politechniki Wrocławskiej. Autor jest adiunktem w Instytucie Informatyki i Elektroniki Uniwersytetu Zielonogórskiego oraz współpracuje z firmami branży informatycznej. Głównymi tematami zainteresowań naukowych są: projektowanie systemów informatycznych, metodyki projektowania używane w inżynierii oprogramowania oraz systemy wielowarstwowe i rozproszone.



e-mail: T.Gratkowski@iie.uz.zgora.pl

Streszczenie

W procesie zarządzania projektem IT, menadżer projektu jest odpowiedzialny za przeniesienie informacji zgromadzonych w wymaganiach na zadania, które zawierają opis czynności do wykonania przez członków zespołu w celu realizacji tworzonego systemu informatycznego. Prowadzone przez autora badania zmierzają do opracowania mechanizmu wspomagającego menadżera w tym procesie, dzięki zastosowaniu dedykowanego repozytorium przechowującego zadania wielokrotnego użycia. W poniższym artykule przedstawiono projekt na podstawie, którego powstała implementacja systemu wspomagającego wielokrotne użycie zadań, będących szablonami postępowania dla członków zespołu. Uzyskany prototyp posłuży do przeprowadzenia dalszych badań eksperymentalnych, zmierzających do opracowania metody pomiaru podobieństwa pomiędzy przechowywanymi w repozytorium zadaniami wielokrotnego użycia a parametrami przekazywanymi w zapytaniu przesyłanym do repozytorium w celu odszukiwania zadań.

Słowa kluczowe: metodyki procesu wytwórczego oprogramowania, wspomaganie procesu wytwórczego, tworzenie listy zadań.

Management platform for reusable tasks

Abstract

A lot of software development methodologies contain stage when a project manager have to create list of tasks for the team. The list is made based on information from use cases (and references to use-case specifications), system-wide requirements, changes and enhancement requests, defects and development tasks. The project manager can use project management tools. This tools helps to manage task list and the manager can use templates of task, but process of creating new task is made manually. This process based on experience and knowledge of a project manager. In this article is presented project of a system to manage reusable tasks. The system will helps the manager to find tasks based on predefined query in Reusable Tasks Repository Query Language (RTRQL) [10]. The reusable task will be stored in dedicated repository. Presented project of system to manage reusable tasks connect result of prior research. The set of functional and non-functional elaborated requirements allow to design architecture of system (Fig. 1). Designed system will be implemented in Service-Oriented Architecture (SOA). It allow to use an implemented prototype through external tools to manage project. The prototype will be used in next steps of research. It helps to find a method of searching reusable task in repository based on similarity of feature and find mechanism of creating query from requirements described in Requirements Specification Language [9].

Keywords: software development methodology, support for task list, support of manage process development.

1. Wprowadzenie

Analizując przykładowe metodyki wytwarzania oprogramowania (ang. software development methodology, SDM) [1], takie jak SCRUM [2], OpenUP [3], RUP [4], PRINCE2 [5] w każdej z nich można wyodrębnić etap, w którym menadżer projektu jest odpowiedzialny za przeniesienie informacji zawartych w wymaganiach funkcjonalnych do zadań, których realizacja przez poszczególnych członków zespołu umożliwi wytworzenie oprogramowania.

Na obecnym etapie wspomagania tego procesu, menadżerowie wykorzystujący oprogramowanie do zarządzania projektami, mogą korzystać z gotowych szablonów zadań, które mogą dopasowywać do potrzeb konkretnego projektu, przypisując przystosowane szablony do realizacji konkretnego wymagania funkcjonalnego. Innym klasycznym sposobem jest zdefiniowanie nowego zadania na podstawie informacji zawartych w wymaganiach funkcjonalnych oraz na podstawie swojego doświadczenia lub członków zespołu uczestniczącego w projekcie.

Prowadzone przez autora badania zmierzają do opracowania metody wspomagania procesu dopasowywania szablonów zadań na podstawie informacji zawartych w wymaganiach funkcjonalnych. Wprowadzenie wspomagania przyspieszy i uprości proces tworzenia zadań niezbędnych do realizacji projektu. W publikacji [6] zaprezentowano ideę bazującą na wykorzystaniu zadań wielokrotnego użycia (ang. Reusable Task, RT) (szablonów zadań), które mogą być zaadaptowane w ramach różnych projektów IT.

W poniższym artykule przedstawiono projekt na podstawie, którego powstała implementacja systemu wspomagającego wielokrotne użycie szablonów zadań, nazywanych dalej zadaniami wielokrotnego użycia. Uzyskany prototyp, posłuży do przeprowadzenia badań eksperymentalnych, których kolejnym głównym celem będzie opracowanie metody wyszukiwania zadań wielokrotnego użycia na podstawie informacji zgromadzonych w wymaganiach funkcjonalnych oraz praktyczna weryfikacja przyjętych założeń.

2. Zadania wielokrotnego użycia

Zadanie realizowane przez poszczególnych członków zespołu może być powiązane z defektem, poprawką, wymaganą zmianą, przypadkiem użycia, scenariuszem przypadku użycia, obsługiwany wymaganie, zapotrzebowaniem klienta oraz każdym innym wymaganie wpływającym na zmianę systemu. Zadania zgrupowane są w listy zadań, które mogą być przypisane poszczególnym etapom lub iteracjom w procesie wytwórczym oprogramowania.

W celu dopasowywania zadań na podstawie zdefiniowanych kryteriów, zadania z wcześniej zrealizowanych projektów oraz nowo przygotowane zadania zostaną zeskalowane w repozytorium zadań wielokrotnego użytku (ang. Reusable Tasks Repository, RTR) [6] jako szablony. Format zadań wielokrotnego użytku zeskalowanych w repozytorium został przedstawiony w pozycji [7]. Zaproponowany mechanizm formalizacji zadań wykorzystuje czterowarstwowy meta-model bazujący na specyfikacji Meta Object Facility (MOF) 2.0 [8]. Zaprojektowany mechanizm formalizacji charakteryzuje się dużą elastycznością zastosowań i może być zaadaptowany do opisu zadań z dowolnej metodyki. Dzięki temu możliwe będzie utworzenie repozytorium składającego zadania, które mogą być następnie dostosowywane do potrzeb dowolnej metodyki oraz narzędzi wspomagających prace menadżerów projektów.

Aby możliwa była próba automatyzacji procesu wyszukiwania niezbędnych informacji z wymagań funkcjonalnych, należało zastosować taki sposób formalizacji zapisu, aby z jednej strony był on czytelny dla uczestników procesu zdobywania wymagań a z drugiej strony na tyle sformalizowany, że będzie on jednoznaczny podczas procesu wyszukiwania niezbędnych informacji. Zadania wielokrotnego użytku przechowywane w repozytorium zostaną ponownie użyte po dopasowaniu ich do wymagań funkcjonalnych projektowanego systemu opisanych przy użyciu języka RSL [9]. W celu wyszukiwania zadań w repozytorium na podstawie kryteriów, opracowano język zapytań repozytorium zadań wielokrotnego użycia (ang. Reusable Tasks Repository Query Language, RTRQL) [10].

3. Architektura systemu

W procesie projektowania prototypu pierwszej kolejności dokonano analizy wymagań projektowych. W tabeli 1 przedstawiono główny problem, który zostanie rozwiązany poprzez zastosowanie projektowanego prototypu.

Tab. 1. Analiza problemu

Tab. 1. Problem analyse

Element	Opis
problem polega na...	skomplikowanym procesie projektowania wielowarstwowych systemów komponentowych
problem dotyczy...	analityków systemowych, projektantów systemów wielowarstwowych
rezultatem problemu jest...	pomijanie fazy projektowania, błędy w projekcie przekreślające możliwość pozytywnego zakończenia projektu
korzyści rozwiązania problemu...	większa dostępność zespołów informatycznych do metody projektowania systemów wielowarstwowych

Na bazie postawionego problemu, opracowano zbiór wymagań funkcjonalnych. Wyodrębniono dwa podstawowe przypadki użycia prototypu:

UC1. Zarządzanie zadaniami wielokrotnego użycia, który zawiera następujące przypadki użycia:

- usunięcie zadania;
- edycja zadania;
- dodanie zadania, które jest realizowane w dwóch wariantach:
 - dodawane jest zupełnie nowe zadanie wielokrotnego użycia
 - nowe zadanie wielokrotnego użycia tworzone jest na podstawie wcześniej utworzonego zadania wykorzystywanego w innym projekcie.

UC2. Wyszukiwanie zadań wielokrotnego użycia, które rozszerzony jest przez dwa warianty wyszukiwania:

- system wyszukuje zadań wielokrotnego użycia na podstawie przesłanego zapytania w języku RTRQL;
- do systemu przesłany jest wiersz ze scenariusza opisu przypadku użycia opisany w języku RSL i na podstawie informacji zawartych w wierszu tworzone będzie zapytanie w języku RTRQL, które posłuży następnie do odnalezienia zadań wielokrotnego użycia.

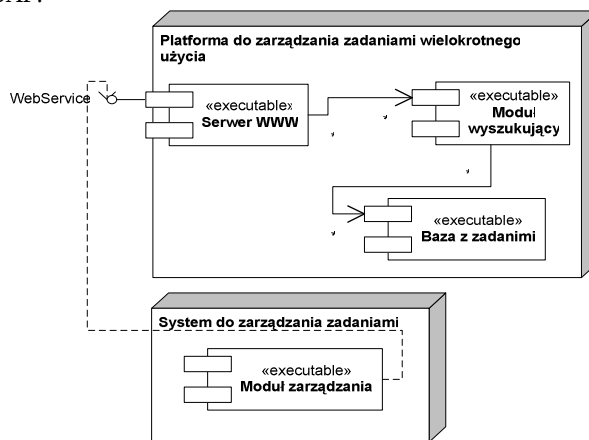
Dodatkowo opracowano zbiór wymagań niefunkcjonalnych, które precyzyjnie wskażą zakres projektowanego systemu:

NF1. System powinien współpracować z zewnętrznymi narzędziami do zarządzania projektem. Dlatego podczas projektowania architektury systemu przyjęto, iż prototyp zostanie zrealizowany zgodnie z architekturą zorientowaną na usługi (ang. service-oriented architecture, SOA) [11]. Wymagać to będzie dodatkowej integracji w każdym z narzędzi zarządzania projektem, w których zostanie wykorzystany prototyp. Zastosowanie SOA przyczyni się do ujednolicenia mechanizmu dostępu do projektowanego systemu.

NF2. Dane zwracane i pobierane przez system powinny być z kodowane jako XML. Zastosowanie architektury SOA pozwoliło na wykorzystanie technologii Web Service z użyciem protokołu SOAP, który stosując protokół HTTP wysła komunikaty w formacie XML. Dzięki temu wymaganiu uproszczony zostanie sposób wymiany informacji pomiędzy heterogenicznymi w tym momencie systemami.

Na rysunku 1 przedstawiono diagram wdrożenia systemu zarządzania zadaniami wielokrotnego użycia. System składa się z dwóch węzłów, klienta – którym jest system do zarządzania projektem, a dokładniej moduł, który pomaga w zarządzaniu zadaniami oraz serwera – który zawiera w sobie trzy artefakty, reprezentujące osobne komponenty odpowiedzialne za poszczególne funkcje systemu. Serwer WWW przetwarza zapytania przychodzące od klientów przy użyciu protokołu SOAP, przekazując zapytanie w RTRQL lub wiersz opisu przypadków użycia zapisanych w RSL do komponentu wyszukującego. Moduł wyszukujący odszukuje odpowiedni zbiór na podstawie zapytania RTRQL lub

przetwarza informacje opisane przy użyciu języka RSL, następnie tworzy zapytanie RTRQL i zwraca odpowiedź do Serwera WWW, które transferuje odpowiedź do klienta z użyciem protokołu SOAP.



Rys. 1. Diagram wdrożenia systemu zarządzania zadaniami wielokrotnego użycia
Fig. 1. Deployment diagram of system to manage reusable tasks

4. Wnioski

W ramach artykułu zaprezentowano projekt platformy do zarządzania zadaniami wielokrotnego użycia. Zapropionowane we wcześniejszych publikacjach metody formalizacji ułatwią proces implementacji. Zdefiniowano główne wymagania funkcjonalne i niefunkcjonalne systemu. Projektowany prototyp będzie ogólnie dostępnym narzędziem świadczącym usługi dla różnych narzędzi wspomagających zarządzanie projektem, dlatego wykorzystano architekturę SOA. Zastosowanie SOA w połączeniu z technologią Web Service i protokołem SOAP pozwoliło na wymianę informacji pomiędzy systemami przy użyciu danych zapisanych w języku XML.

Przedstawiony projekt umożliwił implementację narzędzia, które posłuży do dalszych badań eksperymentalnych. Głównym celem będzie zaprojektowanie mechanizmu wyłuskiwania informacji z wymagań funkcjonalnych w celu tworzenia zapytań w języku RTRQL oraz opracowanie metody pomiaru podobieństwa pomiędzy przechowywanymi zadaniami wielokrotnego użycia a parametrami przekazywanymi w zapytaniu.

5. Literatura

- [1] The Centers for Medicare & Medicaid Services, Selecting a development approach, 27 Oct 2008.
- [2] Pham, A. Pham, P.: Scrum in Action: Agile Software Project Management and Development, Course Technology PTR, January 21 2011.
- [3] http: Introduction to OpenUP (Open Unified Process), Version 4, 30 Nov 2010, <http://epf.eclipse.org/wikis/openup/>, (01.03.2012).
- [4] Kruchten, P.: The Rational Unified Process: An Introduction, Addison-Wesley, 2003.
- [5] Portman, H.: PRINCE2 in Practice, Van Haren Publishing, April 2009
- [6] Gratkowski, T., Wspomaganie procesu definiowania zadań wykonywanych w ramach projektu informatycznego, Metody Informatyki Stosowanej 2010, nr 3, s. 105–109.
- [7] Gratkowski, T.: Formalizacja zadań wielokrotnego użytku, Pomiary, Automatyka, Kontrola - 2011, Vol. 57, nr 7, s. 741–744
- [8] OMG, Meta Object Facility (MOF) Core Specification, Version 2.0, January 2006.
- [9] Requirements Specification Language Definition Defining the ReD-SeeDS Languages Deliverable D2.4.1, version 1.00, 28.02.2007
- [10] Gratkowski, T.: Język zapytań repozytorium zadań wielokrotnego użytku, Integracja systemów informatycznych - nowe wyzwania, Pomorskie Wydaw. Naukowo-Techniczne PWNT, 2011, s. 265–272.
- [11] Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology, and Design, Prentice Hall, August 12, 2005