

UNIWERSYTET ZIELONOGÓRSKI

Wydział Informatyki, Elektrotechniki i Automatyki

Praca dyplomowa

Kierunek: Informatyka

SYSTEM ZARZĄDZANIA PROJEKTAMI DEDYKOWANY
METODYCE SCRUM

Piotr Joński

Promotor:
dr inż. Andrzej Marciniak

Zielona Góra, 02 2016

Piotr Joński

Zielona Góra 10/02/2016

431IDZ

Wydział Informatyki,

Elektrotechniki i Automatyki UZ

OŚWIADCZENIE

Świadomy odpowiedzialności karnej oświadczam, że przedkładana praca dyplomowa pt.

System zarządzania projektami dedykowany metodyce Scrum

została napisana przeze mnie samodzielnie i nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem dyplomu wyższej uczelni lub tytułów zawodowych. Jednocześnie oświadczam, że w/w praca nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994r. o prawie autorskim i prawach pokrewnych innych osób (DZ. U. z roku 2000 Nr 80 poz. 904) oraz dóbr osobistych chronionych prawem cywilnym.

Oświadczam również, że egzemplarz pracy dyplomowej w formie wydruku komputerowego jest zgodny z egzemplarzem pracy dyplomowej w formie elektronicznej.

.....

podpis

Streszczenie

Streszczenie pracy dyplomowej powinno zawierać kilkuzdaniowe omówienie zagadnień poruszanych w pracy. W tej części należy pokrótce scharakteryzować cel oraz podstawowe założenia pracy na tle stanu nauki i techniki w danej dziedzinie, jak również zamieścić od 2 do 6 słów kluczowych.

słowa kluczowe: praca dyplomowa, skład komputerowy, formatowanie dokumentu.

Spis treści

0.1. Historyjki użytkownika	1
---------------------------------------	---

Spis rysunków

Spis tabel

0.1. Historyjki użytkownika

Jednym ze sposobów na prowadzenie dokumentacji projektu, jak i zbioru wymagań jest utrzymywanie rejestru historyjek użytkownika. Historyjki użytkownika są częścią zwinnych metody prowadzenia projektu. Jako że wytwarzanemu systemowi towarzyszy metodyka Scrum, nie mogło tutaj zabraknąć tego elementu.

Historyjka jest jednostką funkcjonalności w projektach XP. Pokazujemy postęp prac, dostarczając przetestowany i zintegrowany kod, który składa się na implementację danej historyjki. Historyjka powinna być zrozumiała i wartościowa dla klientów, testowalna przez programistów i na tyle mała, żeby programiści mogli zaimplementować sześć historyjek w takcie jednej iteracji¹.

Historyjki mogą mieć wiele wzorców. W tej pracy są stosowane dwa z nich:

- Jako *<typ użytkownika>* mogę *<nazwa zadania>*.
- Jako *<typ użytkownika>* mogę *<nazwa zadania>* w celu *<cel>*[6]

W projektowanym systemie występują cztery rodzaje użytkowników (administrator, product owner, scrum master oraz developer) z różnymi uprawnieniami. Teraz szczegółowo zostaną opisane uprawnienia każdego z nich w postaci *user story*. Warto pamiętać, że w systemie występuje uproszczony model użytkowników. Dodatkowo każdy z nich jest developerem, a co za tym idzie, może wykonywać akcje użytkownika developer oraz własne.

Administrator

1. Jako administrator mogę tworzyć nowych użytkowników w celu dodania ich do systemu.
2. Jako administrator mogę usuwać użytkowników z systemu.
3. Jako administrator mogę dodawać użytkowników do grup oraz usuwać z nich w celu modyfikacji zespołu.
4. Jako administrator mogę nadawać dowolne uprawnienia wszystkim użytkownikom.
5. Jako administrator mogę utworzyć projekt w celu dodania go do systemu.

¹K. Beck, M. Fowler, *Planning Extreme Programming*, Addison-Wesley, Boston 2000, s.42

6. Jako administrator mogę usunąć projekt w celu usunięcia z systemu oraz powiązanych z nim elementów t.j. sprint, story, backlog oraz inne. Operacja usuwania odbywa się kaskadowo.
7. Jako administrator mogę modyfikować projekt.
8. Jako administrator mogę tworzyć nowe grupy w celu dodania ich do systemu.
9. Jako administrator mogę dodawać i usuwać grupy z projektów w celu przydzielenia uprawnień.
10. Jako administrator mogę tworzyć, usuwać oraz edytować statusy zadań.
11. Jako administrator mogę tworzyć, usuwać oraz edytować priorytety zadań.

Product owner

1. Jako product owner mogę tworzyć nowe zadania w projekcie, w celu dodania ich do backlogu.
2. Jako product owner mogę nadawać priorytety poszczególnym zadaniom w projekcie.
3. Jako product owner mam wgląd w cały projekt, do którego jestem przypisany.

Scrum master

1. Jako scrum master mogę tworzyć nowe sprinty w projekcie.
2. Jako scrum master mogę tworzyć nowe story w sprintach.
3. Jako scrum master mogę dodawać i usuwać zadania ze story.
4. Jako scrum master mogę tworzyć retrospektywy.
5. Jako scrum master mam wgląd we wszystkie projekty zespołów, w których jestem scrum masterem.

Developer

1. Jako developer mogę tworzyć i edytować zadania.
2. Jako developer mogę dodawać komentarze do zadań oraz retrospektyw.
3. Jako developer mogę usuwać swoje komentarze do zadań oraz retrospektyw.
4. Jako developer mogę edytować swój profil.

Bibliografia

- [1] JavaServer Faces i Eclipse Galileo. Tworzenie aplikacji WWW *Andrzej Marciniak* Helion 2010
- [2] Core JavaServer Faces. Wydanie II *David Geary, Cay S. Horstmann* Helion 2008
- [3] Enterprise JavaBeans 3.0 *Bill Burke & Richard Monson-Haefel* Helion 2007
- [4] JBoss AS 7. Tworzenie aplikacji *Francesco Marchioni* Helion 2014
- [5] Core J2EE. Wzorce projektowe *Deepak Alur, John Crupi, Dan Malks* Helion 2004
- [6] Scrum. O zwinnym zarządzaniu projektami. Wydanie II rozszerzone *Mariusz Chrapko* Helion 2015
- [7] Java. Kompendium programisty. Wydanie VIII *Herbert Schildt* Helion 2012
- [8] Zwinne wytwarzanie oprogramowania. Najlepsze zasady, wzorce i praktyki *Robert C. Martin* Helion 2015
- [9] Git. Rozproszony system kontroli wersji *Włodzimierz Gajda* Helion 2013
- [10] Mistrz czystego kodu. Kodeks postępowania profesjonalnych programistów *Robert C. Martin* Helion 2013
- [11] Czysty kod. Podręcznik dobrego programisty *Robert C. Martin* Helion 2010
- [12] Rusz głową! Wzorce projektowe *Eric Freeman, Elisabeth Freeman, Bert Bates, Kathy Sierra* Helion 2011
- [13] Refaktoryzacja do wzorców projektowych *Joshua Kerievsky* Helion 2005