

УДК 004.5

## ХМАРНИЙ СЕРВІС ДЛЯ УПРАВЛІННЯ РОЗРОБКОЮ ПРОГРАМНИХ ДОДАТКІВ НА БАЗІ СЕРВЕРА REDMINE

М. С. Пасека

*Івано-Франківський національний технічний університет нафти і газу,  
вул. Карпатська, 15, Івано-Франківськ, 76000, Україна*

*Проаналізовано доцільність використання сервера Redmine як хмарного сервісу для управління розробкою прикладних програмних додатків як для незначних проектних розробок, так і для навчальних цілей. Окреслено особливості розгортання хмарного сервісу для використання у спільній роботі над прикладними програмними проектами. Надано фахові рекомендації зі встановлення цього сервера та його конфігурування для групової роботи, а також висвітлено певні зауваження щодо роботи із сервером Redmine під час адміністрування одного або декількох програмних проектів одночасно. А також визначено клас та обсяг цих задач де можна використовувати цей хмарний сервіс.*

***Ключові слова:** інформатизація освіти, розробка програм, управління проектами, програмні задачі, хмарні сервіси, прикладні програмні додатки*

**Постановка проблеми.** Організація кооперативної (групової) роботи у проектному менеджменті розробки прикладних програмних додатків. Redmine є відкритим серверним Web-додатком для управління проектними програмними розробками та завданнями, а також використовується для моніторингу і відстеження помилок. Прикладний додаток Redmine написаний мовою програмування Ruby і є додатком на основі відомого веб-фреймворку Ruby on Rails.

Функціональні можливості прикладного додатку Redmine: ведення декількох незалежних програмних проектів; гнучка система колективного доступу, заснована на ролях; система фіксації та відстеження помилок; ведення діаграм Ганта та щогодинний календар виконання робіт; відстеження планових та нових завдань з програмного проекту, а також документів і управління файлами; миттєве оповіщення всіх зацікавлених осіб про корегування програмного проекту за допомогою RSS-потоків та електронної пошти; організація кооперативних форумів для кожного програмного проекту; ведення та фіксація обліку тимчасових витрат із розробки прикладного програмного додатку; можливість додаткового налаштування довільних полів для інцидентів, а також тимчасових витрат на проект і користувачів; забезпечення легкої інтеграції із системами управління різними версіями (Mercurial, SVN, Bazaar, CVS, Git, і Darcs); створення журналу записів про помилки та відхилення від проектних планів на основі отриманих повідомлень; забезпечення множинної підтримки автентифікації LDAP; можливість самостійного корегування персональних даних користувачів та реєстрації нових; підтримка

багатомовного інтерфейсу; забезпечує інтеграцію системам керування базами даних Microsoft SQL Server, MySQL, PostgreSQL, SQLite, Oracle.

**Аналіз остатніх досліджень та публікацій.** Кооперативну роботу управління під час роботи над програмними проектами можна розглядати як систематичний процес створення, зберігання та розподілу людських і технічних ресурсів. У науковій публікації розглядають актуальні питання з управління розробкою програмних додатків у розрізі глобальної програмної інженерії та особливості командної співпраці всіх учасників проекту [1, 2]. Методології управління над програмними проектами сьогодні трансформуються у певні стандарти: «Стандарт з організації компетенцій» [3–6], «Стандарт з оцінки ефективної діяльності у проектах» [7–9], «Стандарт за індивідуальними компетенціями для управління проектами, програмами та портфелями проектів» [10–12]. Сьогодні немає ефективної структури знань про методологію розробки управління проектами програмних додатків проте вдосконалюються міжнародні стандарти «P16326:201x WD5, Dec 2017 - ISO/IEC/IEEE International Standard - Systems and Software Engineering--Life Cycle Processes--Project Management» IEEE 12207 та ISO/MEK, IEEE 15288:2015, цей стандарт використовують для менеджерів програмних проектів з надання фахової підтримки управління для успішного завершення прикладних програмних проектів. Цей міжнародний стандарт розширює ґрунтовне керівництво проектом для успішного управління і маніпулювання технічними процесами під час розробки прикладних програмних продуктів [13, 14].

**Мета статті** — використання сервера Redmine в середовищі ОС Windows як безкоштовного хмарного сервісу, який забезпечить груповий менеджмент управління та швидку реакцію на зміни як внутрішнього, так і зовнішнього оточення під час розробки прикладного програмного додатку і стає критичним чинником успіху цього проекту.

**Виклад основного матеріалу дослідження.**

### Структура бази даних сервера Redmine

Принципи	Коротка характеристика принципів
<b>Користувачі системи</b>	Користувачі сервера Redmine є одним із основних понять предметної області. Модель функціонування — основа для ідентифікації та аутентифікації користувача, що працює з системою персоналу із клієнтів, а також для розподілення їх в різних ролях проекту.
<b>Ролі</b>	Ролі користувачів визначаються адаптивною моделлю визначення прав доступу користувачів. Ролі містять набір прав та привілеїв, що дозволяють розмежовувати доступ до різноманітних функціональностей системи. Користувачам призначають роль у кожному проекті окремо, в якому він бере участь. Користувач може мати декілька ролей, а призначення ролі для окремого завдання (issue) в цей момент неможливо.
<b>Проекти</b>	Проект є одним з основних понять у цій предметній сфері, а саме — систем управління проектами. Завдяки цьому поняттю можна налагодити кооперативну роботу і планування людських та фінансових ресурсів в декількох проектах одночасно з розмежуванням рівня прав доступу різним користувачам. Проекти допускають ієрархічну вкладеність завдань та задач.

<b>Трекери</b>	Трекери — базова класифікація, за якою впорядковуються завдання в програмному проекті. Поняття «трекер» входить до систем обліку помилок (англ. Bug tracking tool), в якому відображено кожний проект окремо. Отже, в «Redmine» трекери — аналог підкласів класу «Завдання» та є основою для поліморфізму різних завдань, а також дають змогу визначати різні типи полів. Прикладами трекерів є: «Помилка», «Поліпшення», «Підтримка», «Документування».
<b>Завдання</b>	Завдання є базовим поняттям всієї системи управління проектами, що описують поставлену задачу, яку необхідно виконати. У кожному завданні (task) є такі атрибути, як опис і автор та обов'язково завдання має бути прив'язане до трекера. Кожне завдання набуває певного статусу. Статус — це окрема сутність з можливістю визначення прав та привілеїв для різних ролей. Для кожного програмного проекту необхідно окремо визначити набір етапів (послідовності) розробки та набір категорій завдань.
<b>Відстеження зміни статусу завдань</b>	За контролем корегування параметрів завдань проекту користувачами у системі сервера Redmine відповідають дві сутності: «Змінений параметр» та «Запис журналу змін». Сутність «Змінений параметр» відображається в окремому журналі та призначений для фіксації старого і набутого значення користувачем параметра. Запис у журналі відображає лише одну дію користувача з корегування параметрів завдання або додавання коментаря до нього. Отже, запис може слугувати одночасно як інструментом ведення історії завдання проекту, так і інструментом ведення діалогу.
<b>Зв'язки між завданнями</b>	Задачі можуть бути пов'язані між собою, наприклад, одна задача може мати декілька вкладених задач або передувати іншій задачі. Ця інформація використовується у процесі планування розробки прикладних програмних додатків, а за її збереження в Redmine відповідає окрема сутність.
<b>Облік витраченого часу на проект</b>	В Redmine є система обліку витраченого часу, що контролює «Витрачений час», який пов'язаний з виконавцями та задачами. Ця сутність дає змогу контролювати витрачений час на діяльність виконавця (розробка, проектування, підтримка) і додавати короткі коментарі до задачі. Цю інформацію можна використати для контролю затраченого часу і оплати його.
<b>Прив'язка репозиторіїв</b>	Сервер Redmine дає змогу ефективної та оперативної інтеграції з різноманітними прикладними програмами контролю версій (репозиторіями). Інтеграція та моніторинг дають змогу відстежувати зміни у зовнішньому репозиторії та фіксацію змін у базі даних, проводити аналіз змін у проектній розробці з метою контролю їх та прив'язки до завдань. У інфологічній моделі структури бази даних системи за кооперацією із зовнішнім репозиторієм відповідають три сутності, а саме: «Сховище», «Редакція» і «Зміна».
<b>Отримання повідомлень</b>	Отримання повідомлень користувачем про зміни, які відбулися у проектній розробці, відбуваються у сутності «Спостерегачі», що зв'язує виконавців з об'єктами різноманітних класів (проекти, задача, завдання, форуми та ін.).

## Розгортання хмарного сервісу сервера Redmine для управління проектом програмної розробки

Для завантаження хмарного сервера Redmine необхідно встановити програмне середовище Ruby. На рис.1 відображено моніторинг встановлення Ruby. У процесі встановлення активізуємо прапорці на «Add Ruby executables to your PATH» та «Associate .rb and .rbw files with this Ruby installation».

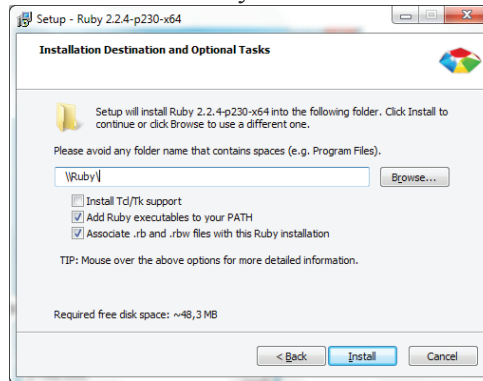
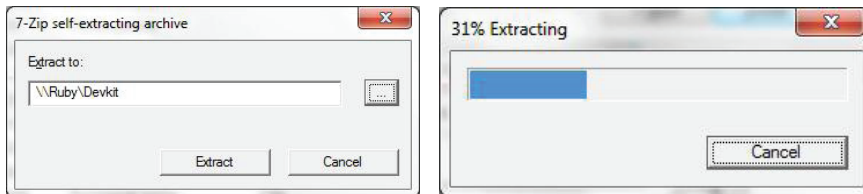


Рис. 1. Моніторинг встановлення Ruby

Також для розгортання хмарного сервера Redmine необхідно встановити Ruby DevKit 4.7.2. У консолі переходимо в папку з DevKit і виконуємо команду `ruby dk.rb init` і `ruby dk.rb install` (рис. 2). Завантажуємо Redmine на хмарний сервіс та розархівуємо його в папці де встановлений Ruby, а також копіюємо сертифікат безпеки з веб-сторінки у робочий файл `\\Ruby\lib\ruby\2.1.0\rubygems\ssl_certs`.



а)

б)

```

\\Ruby\DevKit>ruby dk.rb init
Initialization complete! Please review and modify the auto-generated
' config.yml ' file to ensure it contains the root directories to all of the
installed Rubies you want enhanced by the DevKit.
\\Ruby\DevKit>ruby dk.rb install
[INFO] Skipping existing gen override for ' \\Ruby '
[WARN] Skipping existing DevKit helper library for ' \\Ruby '

```

в)

Рис. 2. Моніторинг встановлення Ruby DevKit

Наступним етапом розгортання хмарного сервера Redmine є встановлення менеджера керування «гемами» командою «`gem install bundler`» та SQLite для мови програмування Ruby «`gem install sqlite3-ruby`». Встановлюємо полегшену версію керування реляційними базами даних на основі мови запитів SQL. На наступному кроці оновлюємо всі «геми» командою «`gem update`» (рис. 3).

```
\\Ruby\DevKit>gem install bundler
Fetching: bundler-1.11.2.gem <100%>
Successfully installed bundler-1.11.2
Parsing documentation for bundler-1.11.2
Installing ri documentation for bundler-1.11.2
Done installing documentation for bundler after 16 seconds
1 gem installed
```

а)

```
\\Ruby\DevKit>gem install sqlite3-ruby
Fetching: sqlite3-1.3.11-x64-mingw32.gem <100%>
Successfully installed sqlite3-1.3.11-x64-mingw32.gem
Fetching: sqlite3-ruby-1.3.11.gem <100%>
```

б)

```
\\Ruby\DevKit>gem update
Updating installed gems
```

в)

Рис. 3. Оновлення «gem» командою «gem update»

Для налаштування конфігурації парного сервера Redmine потрібно провести його конфігурування, а саме: перейти у каталог config відкрити файл database.yml.example для редагування, видалити усі записи, крім тих, які стосуються SQL3 і зберігаємо під новою назвою database.yml. В кореновому каталозі Redmine відкриваємо конфігураційний файл Gemfile, у якому на початку прописуємо «gem «sqlite3-ruby»». Наступним кроком налаштування хмарного сервера Redmine необхідно перейти у кореневу папку та виконати команду «bundle install» (рис. 4).

```
\\Ruby\redmine>bundler install
Using rake 11.0.1
Using i18n 0.6.11
Using multi_json 1.11.2
Using builder 3.0.4
Using erubis 2.7.0
Using journey 1.0.4
Using rack 1.4.7
Using hike 1.2.3
Using tilt 1.4.1
Using mine-types 1.25.1
Using polyglot 0.3.5
Using arel 3.0.3
Using tzinfo 0.3.46
Using coderay 1.1.1
Using htmlentities 4.3.1
Using json 1.8.3
Using thor 0.19.1
Using net-ldap 0.3.1
Using rails 3.2.22
Bundle complete! 28 Gemfile dependencies, 41 gems now installed.
Gems in the groups development, test and rmagick were not installed.
Use 'bundle show [gemname]' to see where a bundled gem is
installed.5666666666xxx
```

Рис. 4. Консольне вікно інсталяції «bundler»

Наступним кроком виконуємо команду «rake generate\_secret\_token» (рис. 5).

```
\\Ruby\redmine>rake generate_secret_token
\\Ruby\lib\ruby\gems\2.2.0\gems\htmlentities-
4.3.1\lib\htmlentities\mappings\expanded.rb:465 warning: duplicated
key at line 466 ignored: "inodot"
```

Рис. 5. Консольне вікно виконання команди «rake generate\_secret\_token»

Також для встановлення хмарного сервера Redmine необхідно виконати команди «rake redmine:load\_default\_data RAILS\_ENV=>production» та команду «ruby script/rails s -e «production»» (рис. 6).

```
\\Ruby\redmine>rake redmine: load_default_data RAILS_ENV=
production
\\Ruby\lib\ruby\gems\2.2.0\gems\htmlentities-
4.3.1\lib\htmlentities\mappings\expanded.rb:465 warning: duplicated key
at line 466 ignored: "inodot"
Select language: ar, az, bg, bs, ca, cs, da, de, el, en, en-GB, es, et, Eu.
Fa, fi, fr, gl, he, hr, hu, id, it, ja, ko, lt, lv, mk, mn, nl, no, pl, pt, pt-BR,
ro, ru, sk, sl, sq, sr, sr-YU, sv, th, tr, uk, vi, zh, zh-TW, [tn] uk
```

а)

```
\\Ruby\redmine>ruby script/rails s -e "production"
\\Ruby\lib\ruby\gems\2.2.0\gems\htmlentities-
4.3.1\lib\htmlentities\mappings\expanded.rb:465 warning: duplicated key
at line 466 ignored: "inodot"
=> Booting WEBrick
=> Rails 3.2.22 application starting in production on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
```

б)

Для професійного використання хмарного сервера Redmine необхідно встановити та налаштувати плагін Backlogs (рис. 7):

```
\\Ruby\lib\ruby\gems\2.2.0\gems\htmlentities-
4.3.1\lib\htmlentities\mappings\expanded.rb:465 warning: duplicated key at
line 466 ignored: "inodot" 2.6.9.stable. You are running backlogs v1.0.6,
latest version is 1.0.6

=====
Redmine Backlogs Installer
=====

Installing to the production environment.
Fetching card labe is from http://git.gnome.org...done!
Configuring story and task trackers...

-----
Which trackers do you want to use for your stories?
  1. Помилка
  2. Властивість
  3. Підтримка
Separate values with a space <e.g. 1 3>: 1 2 3
You selected the following trackers: Помилка, Властивість, Підтримка.
Is this correct? <y/n> y

-----
Creating a new task tracker.
Please type the tracker's name: tracker1
You type 'tracker1'. Is this correct? <y/n> y
Story and task trackers are now set.
Migrating the database...
```

Рис. 7. Консольне вікно виконання команд встановлення плагіну Backlogs

- розпаковуємо папку з плагіном «`\\ruby/redmine/plugins/`»;
- запускаємо на виконання команду «`set RAILS_ENV=production`»;
- виконуємо інсталяцію плагіну командою «`bundle exec rake redmine:backlogs:install`»

У результаті проведених операцій встановлення хмарного сервера Redmine для проектного управління та моніторингу розробки прикладних програмних додатків отримаємо безкоштовний Web- сервіс (рис. 8).

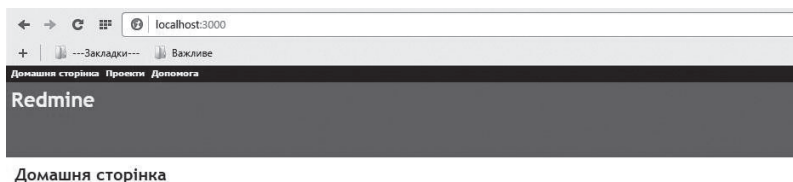


Рис. 8. Робоче вікно хмарного сервера Redmine

### *Деякі зауваження щодо особливостей використання сервера Redmine як хмарного web-додатку:*

- керування документами та файлами у сервері Redmine зводиться до операцій над ними, а саме: додавання, видалення і редагування їх. Проте правами доступу та маніпулюваннями над ними не можна;
- у системі оповіщення сервера Redmine немає повідомлення про маніпулювання з документами;
- у сервері Redmine можна управляти лише правами доступу до завдань програмного проекту лише на рівні окремих інформаційних полів. Наприклад, зараз момент немає змоги приховувати оцінку робочого часу над завданнями програмного проекту від клієнтів. Проте є потенційна можливість зробити додаткові поля видимими лише користувачам з певними про файлами;
- у сервері Redmine немає такої можливості, як визначення коректної послідовності дій у workflow (що необхідно зробити змінюючи статус певних дій). Наприклад, зараз немає змоги поставити вказівник про те, що коли тестувальник закінчив виправляти помилку в програмному taskу. Отже програм менеджер має вибрати відповідального тестувальника та прикріпити до тестувальника номер необхідного білда. Також немає змоги приховати внутрішню переписку в межах програмного проекту між учасниками групи розробників від клієнта.
- у сервері Redmine у списку завдань немає змоги задати загальну трудомісткість окремих задач програмного проекту;
- немає змоги задати користувачеві глобальну роль для всієї системи, наприклад, «Керівник проектної офісу», «Портфельний керівник офісу», які повинні мати можливість доступ до усіх програмних проектів, що є зареєстровані у системі, для цього потрібно виконати монотонну роботу, а саме — додати відповідного користувача із роллю «Керівник проекту».

Можна виділити ще інші проблеми у сервері Redmine: незручності під час управління програмними завданнями із великою кількістю taskів (завдань),

переміщенням цих завдань між різними проектами, пошуком в історії модифікацій завдань, фаховим налаштування типових оповіщень по e-mail тощо), проте ці проблеми не є критичними, їх легко виправити і наймовірніше така функціональність з'явиться у сервері Redmine протягом найближчих релізів. А проблеми з про файлами та правами доступу і підтримкою ієрархічних зав'язків є наслідком недоопрацьованої архітектури сервера Redmine, боротися з такими проблемами функціональності вже дуже складно. На нашу думку, сервер Redmine є відмінним вибором сервісної програми для невеликих проектів і команд та у навчальних цілях, а для великих проектів варто використовувати комерційні системи управління розробки програмних проектів.

**Висновки.** Проаналізувавши необхідність в on-line сервісах управління проектною розробкою прикладних програмних додатків, ми зупинились на розгортанні хмарного сервісу, а саме — сервера Redmine. Визначили особливості розгортання останнього для використання у спільній роботі над прикладними програмними проектами. Надали фахові рекомендації із встановлення цього сервера та його конфігурування для групової роботи, а також висвітлили певні зауваження щодо роботи із сервером Redmine під час адміністрування одного або декількох програмних проектів одночасно.

#### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bushuyev Sergey D., Wagner Reinhard F. IPMA Delta and IPMA Organisational Competence Baseline (OCB): New approaches in the field of project management maturity. *International Journal of Managing Projects in Business*. 2014. Vol. 7. Iss: 2. Pp. 302–310.
2. Convergence of knowledge in project management Bushuyev S., Bushuyev D., Rogozina V., Mikhieieva O. *Proceedings of the 2015. IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2015. 2*. Pp. 496–500.
3. Dafoulas G. Understanding Collaboration in Global Software Engineering (GSE) Teams with the Use of Sensors: Introducing a Multi-sensor Setting for Observing Social and Human Aspects in Project Management / Georgios Dafoulas, Cristiano Maia, Almaas Ali, Juan Carlos Augusto, Victor Lopez-Cabrera. 2017 *International Conference on Intelligent Environments (IE)*. 2017. Pp. 114–121.
4. Individual Competence Baseline for Project, Programme & Portfolio Management, Version 4. (2015). *International Project Management Association*, 415 p.
5. IPMA Organisational Competence Baseline (IPMA OCB) (2013). IPMA, 2013. 67 p.
6. ISO/IEC/IEEE International Standard - Systems and Software Engineering--Life Cycle Processes--Project Management ISO/IEC/IEEE P16326:201x WD5, December 2017. Pp. 1–29.
7. Kerzner H. *Strategic Planning for Project Management Using a Project Management Maturity Model* (2001). John Wiley & Sons Inc. 255 p.
8. *Managing Successful Programmes*, Office of Government Commerce (2007). UK, 258 p.
9. Qian Li-Ping, Yang Xiao-Ping, Song Yu. Design for Ontology Knowledge Base Based on Structural Members. *International Journal of Database Theory and Application*. 2015. Vol. 8. No. 5. Pp. 27–38.



10. Project success analysis framework: A knowledge-based approach in project management / Todorović M., Petrović D., Mihić M., Obradović V., Bushuyev S. *International Journal of Project Management*. Volume 33. Issue 4. Pp. 772–783.
11. Креативные технологии управления проектами и программами / С. Д. Бушуев, Н. С. Бушуева, И. А. Бабаев, В. Б. Яковенко, Е. В. Гриша, С. В. Дзюба, А. С. Войтенко. Київ : Саммит–Книга, 2010. 768 с.
12. Бушуев С. Д., Харитонов Д. А., Рогозина В. Б. Организационные патологии управления проектами. *Управління розвитком складних систем*. 2012. № 10. С. 5–8.
13. Неизвестный С. *Мозг проекта*. Москва : Российские научное издательство, 2007. 400 с.
14. Ярошенко Ф. А., Бушуев С. Д., Танака Х. *Руководство инновационными проектами и программами на основе системы знаний P2M*. Киев : Саммит-Книга, 2012. 272 с.

### CLLOUD SERVICE FOR MANAGING THE APPLICATION DEVELOPMENT ON THE REDMINE SERVER BASIS

M. S. Pasięka

*Ivano-Frankivsk National Technical University of Oil and Gas,  
15, Karpatska St., Ivano-Frankivsk, 76019, Ukraine*

*This paper analyzes the feasibility of using the Redmine server as cloud service to manage the development of software applications for minor projects and for training purposes. The features of deploying a cloud service for use in the collaborative work over applied software projects have been outlined. The professional advice as to the installation of this server and its configuration for the group work has been given and some observations on working with the Redmine server in the administration of one or multiple software projects simultaneously have been highlighted. Also the class and the tasks involved, where this cloud service can be used, have been identified.*

**Keywords:** *informatization of education, software development, project management, software task, cloud services, applied software application.*

*Стаття надійшла до редакції 00.00.2018.*