Виконали студенти групи 2153ст:

Мамонова Зоряна

Фанькіна Алла

Тимошин Георгій

Білий Владислав

Комаровський Іван

Гриненко Олег

Мета роботи: придбання практичних навичок командної роботи з використанням різних систем комунікацій.

Завдання: скласти опис до декількох систем комунікацій, проаналізувати надані можливості та ефективність даних систем, обґрунтувати вибір тієї або іншої системи комунікацій, створити проект, зареєструвати в ньому команду проекту, створити спринти, занести туди роботи, розподілити відповідальність між працівниками, виконати перший спринт, продемонструвати спринт-беклог.

В якості систем комунікацій можуть бути використані системи відслідковування помилок (Jira, Mantis, Redmine або інші), та системи обліку заявок.Теоретичні відомості

Використання методології управління проектами передбачає створення спеціальної групи (команди проекту), яка стає самостійним учасником проекту і здійснює управління інвестиційним процесом в рамках проекту. Ця група створюється на період реалізації проекту і після його завершення розпускається.

Гнучкі методології розробки (Agile software development) ‑ це концептуальний каркас, в рамках якого виконується розробка програмного забезпечення.

Agile ‑ методи роблять упор на безпосереднє спілкування лицем до лиця. Більшість agile ‑ команд розташовані в одному офісі, іноді званому англ. bullpen . Як мінімум, вона включає і « замовників » (англ. product owner ‑ замовник чи його повноважний представник, що визначає вимоги до продукту ; цю роль може виконувати менеджер проекту, бізнес ‑ аналітик або клієнт). Офіс може також включати тестувальників, дизайнерів інтерфейсу, технічних письменників і менеджерів.

Основний метрикою agile ‑ методів є робочий продукт. Віддаючи перевагу безпосередньому спілкуванню, agile ‑ методи зменшують обсяг письмової документації в порівнянні з іншими методами. Це призвело до критики цих методів як недисциплінованих.

Хід роботи

Порівняння GitHub і BitBucket

Загальні риси:

1. Пропонують безкоштовний Git-хостинг для відкритого коду, а також можливість створювати wiki і багтрекер;
2. Елементи соціальної мережі (легке створення ФОРКІВ і т п).
3. Є підтримка HTTPS;
4. Є власні API
5. Підтримується Gravatar;
6. Наявність інтеграції з Twitter, FriendFeed та іншими сервісами;
7. У GitHub і BitBucket є блоги, а також мікроблоги в Twitter;
8. Написано чимало мобільних додатків для обох сервісів;

Негативні риси двох сервісів:

1. Багтрекер не дозволяє пов'язувати залежні один від одного так, як це вміє Jira.
2. Немає локалізації. На GitHub була, та видалили.
3. У обох сервісів негідні embed-коди (не завжди і не скрізь коректно відображаються), на pastebin набагато краще;
4. Обидва, трапляється, лежать;

Переваги GitHub:

1. Є Gist, підтримка якого у свій час навіть була в твіттері, але зараз її, видалили;
2. Веб-інтерфейс дозволяє подивитися «git blame»;
3. Є інтеграція з IRC, Jabber, Jira (дивно, що немає на BitBucket), MantisBT, Bugzilla, підтримка Post-Receive URLs і багато чого ще, притому всі ці компоненти - OpenSource;
4. Свій скорочувач посилань git.io, правда у неї немає нормального користувальницького інтерфейсу;
5. Є підтримка Subversion, яка в даний час доопрацьовується. У бітбакета вона була, але в даний час закинута;
6. Більш просунутий пошук. Наприклад, можна шукати програмістів по країні і використовуваному мови програмування. Розробники бітбакета кажуть, що вже працюють над поліпшенням пошуку;
7. Малює красиві картинки (графіки активності, «Stats & Graphs» при перегляді репозиторію);

Переваги BitBucket:

1. Належить відомій Atlassian (Jira, Confluence);
2. Підтримує безкоштовні закриті репозиторії. Причому у закритого репозиторію може бути, скажімо, відкритий багтрекер або навпаки. На гітхабі теж є закриті репозиторії, але не за безкоштовно;
3. Є інтеграція з Akismet, Google Analytics і не тільки;
4. Єдине обмеження у безкоштовних акаунтів є в тому, що доступ до закритого репозиторію можуть мати не більше 5-и осіб. Жодного обмеження за обсягом використовуваного дискового простору немає. На гітхабі за безкоштовно дають тільки 300 Мб;
5. Крім Git також підтримується Mercurial. У цієї VCS є свої цікаві особливості.
6. На наш погляд, інтерфейс у бітбакета продуманий краще, ніж у гітхаба. Наприклад, з будь-якої сторінки можна з легкістю перейти в один зі своїх сховищ;
7. Є імпорт Git / Mercurial / Subversion репозиторіїв з GitHub і SourceForge. Зручно для перенесення своїх проектів або форків чужих;
8. Можна розсилати e-mail інвайт в свій проект;
9. Обліковий запис можна прив'язати до власного домену.
10. Нормальна техпідтримка на groups.google.com.
11. Підтримується вхід по OpenID.

Проаналізувавши дві системи, вибір нашої команди впав на систему GitHub, завдяки її зручності (в деяких аспектах), а також завдяки необхідності даної системи, як обов’язкового навику при працевлаштуванні.

Спринт беклог представлений на рисунках 1-3

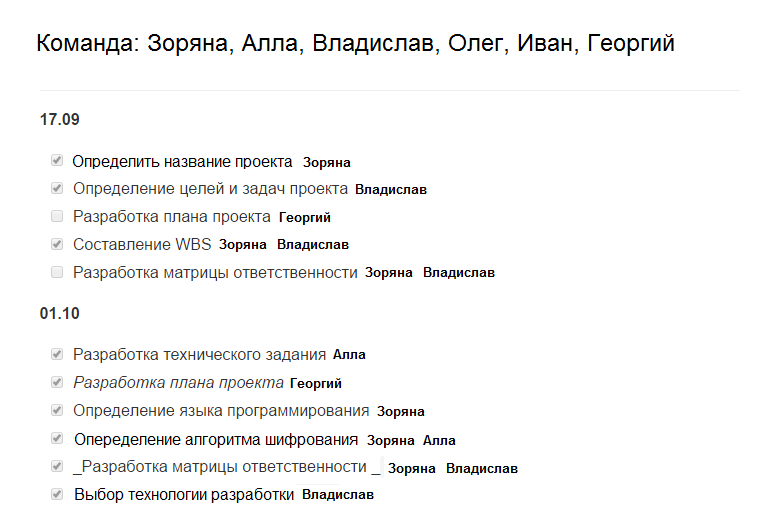


Рисунок 1 Sprint Backlog за перший місяць

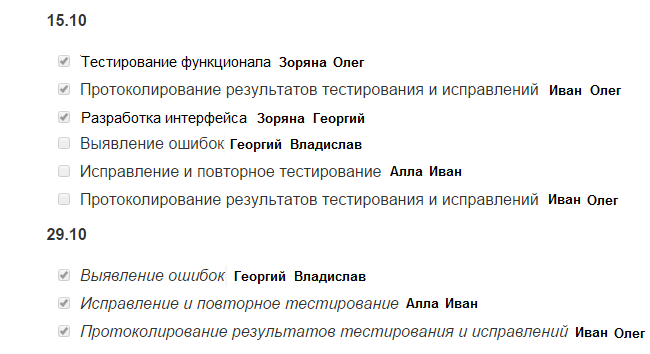


Рисунок 2 Sprint Backlog за другий місяць

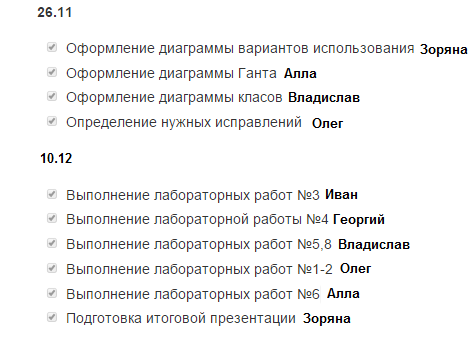


Рисунок 3 Останній Spint Backlog

Висновок: порівняли існуючі системи комунікації, зробили вибір на GitHub, створили Sprint Backlog.