

# Лабораторна робота 3

Проектування інформаційних систем

lakub.muravlov@gmail.com [Switch accounts](#)

 Draft restored

The name and photo associated with your Google Account will be recorded when you upload files and submit this form. Your email address is not part of your response.

**\*Required**

Ім'я \*

Андрій

Прізвище \*

Муравльов

Група \*

☐ ДА-81

☒ ДА-82

Я не списую з будь який джерел, не користуюся лабораторними роботами своїх колег, не роблю копії відповідей з інших лабораторних робіт і т.д. \*

☒ Так



**Мета роботи:** Вивчити типові життєві цикли розробки програмного забезпечення. Усвідомити методику роботи з багтрекінгом, навчитися створювати питання, задавати їх статус, а також навчитися оперувати статистикою багтрекінга.

**Задача:** Вивчити зміст прикладу інформаційної системи та створити опис життєвого циклу розробки відповідного програмного забезпечення. Використовуючи систему багтрекінга MantisBT або подібну необхідно ознайомитися веденням журналу для повного циклу розробки та тестування, відтворити статуси для питання (ticket) та ролі у процесі розробки програмного забезпечення.

Опишіть що таке є SDLC ? \*

SDLC - жизненный цикл разработки программного обеспечения. SDLC описывается методологией принятой для имплементации продукта программного обеспечения. Парадигма жизненного цикла определяет общую организацию продукта, его основные фазы и переход между ними.

Жизненный цикл ПО сопровождается разработкой, оборотом и использованием программной документации



Який життєвий цикл має кожна задача в системі багтрекінгу ? \*

Каждая задача в системе багтрекинга имеет следующие этапы:

- New

Тестирующий обнаружил баг, описал шаги чтобы его воспроизвести, и занес тикет в систему багтрекинга.

- Opened

После отправки тестировщиком ошибки она или автоматически через пайплайны, или вручную назначается на человека который должен эту ошибку проанализировать, обычно этим человеком является Project Manager. После рассмотрения менеджером ошибки тикет может получить следующие значения:

- Deferred

Исправление ошибки не несет ценности на данном этапе разработки или его имплементация отнимет слишком много времени и ресурсов, и есть более приоритетные задачи.

- Rejected

По причине неактуальности отклоняется

- Duplicate

Описанная тестировщиком ошибка уже была внесена в систему багтрекинга.

- Assigned

Ошибка актуальна и должна исправлена в следующем билде. Менеджер назначает тикет на разработчика который должен исправить ошибку.

- Fixed

Разработчик ответственный за исправление ошибки заявляет об устранении бага.

-Verified

Тестирующий проверяет действительно ли дефект исправлен. Если баг отсутствует тикет приобретает этот статус.

- Reopened

Если баг не исправлен или появился спустя некоторое время после закрытия тогда тикет приобретает этот статус.

- Closed

В следствии некоторого количества таких циклов баг окончательно устранен

Чим відрізняється SDLC від життєвого циклу задачі? \*

SDLC определяет процес разработки ПО, его фазы, методологию и парадигму. В то время как жизненный цикл задачи имеет отношение исключительно к процессу тестирования, этапы которого имеют связь исключительно с тестированием и выявлением ошибок, а не разработкой в общем.



Намалюйте діаграму життєвого циклу вашої задачі з урахуванням ролей учасників вашого проекту. \*



ticket-diagram.png X

Приєднайте файл зі скріншотами роботи з задачами згідно життєвого циклу вашої задачі. Використовуючи систему багтрекінга необхідно виконати наступні дії: створити питання (ticket), видалити питання, змінити статус, додати коментар, прикріпити файл, ознайомитися з журналом змін, ознайомитися з виведенням статистики. \*



Lab3.pdf X

### Контрольні питання

Опишіть ролі керівника проекту, архітектора, програміста і тестувальника і т.д. вмаршруті проходження питання. \*

Тестировщик находит баг, создает тикет с его описанием, шагами для воспроизведения; подтверждает исправление бага и закрывает тикет или переоткрывает его.

Проджект менеджер/тимлід; назначает изначальный статус ошибки и передает ее разработчикам на исправление

Разработчик: исправляет ошибку и передает ее тестировщику на рассмотрение.



Опишіть життєвий цикл питання (ticket) у системі багтрекінгу? Як це пов'язано з життєвим циклом виправлення помилок в процесі розробки ПО?

\*

Новый - статус назначен автоматически после внесения баг репорта

Открытый - проведена валидация тимлидом/ПМом и ошибка является актуальной

Исправлен - ошибка устранена разработчиком

Повторно открыта - ошибка не устранена или появилась снова

Закрыта - баг пофикшен и больше не репродюсится

Это самый простой жизненный цикл тикета. Все шги из него напрямую связаны с исправлением ошибок в процессе разработки, но кроме основных статусов в разработке ПО могут использоваться дополнительные:

- назначен
- требуется дополнительная информация
- в процессе исправления
- не репродюсится
- отправлено на юнит тесты/регрессионное тестирование

Чи дозволяє система багтрекінгу проводити експертний аналіз програмного коду алгоритмічних мов програмування ООП? \*

☒ ТАК

☐ Ні



Висновки до лабораторної роботи. Опишіть, які були проблеми при виконанні лабораторної і як ви їх вирішили \*

В процессе выполнения лабораторной работы ознакомился с понятиями ситемы багтрекинга, жизненного цикла ПО и тикета. Ознакомился с системой багтрекинга MantisBT и создал в ней сет ролей по которым прогнал жизненный цикл одного тикета.

Из проблем с выполнением лабораторной работы можно отметить само взаимодействие с мантисом, поскольку он требует дополнительной настройки среды (накатить php и mysql), кроме того запросы на изменение статуса тикета обрабатываются программой достаточно долго. Из альтернатив можно использовать Jira, Trello, корпоративный пакет Notion. Все они имеют функционал необходимый для системы багтрекинга и являются более функциональными и менее ресурсоемкими.

Submit

Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#).

Google Forms

