

- тустувальник) повинен одразу про це дізнатися, щоб працювати з релевантною версією. Також, якщо додаються\скасовують вимоги – команда повинна як умого швидше дізнатися про це;
- д) кастомні WF необхідні у нестандартних проектах, де може знадобитися більш детальне відстеження стану кожного окремого елемента програмного продукту. То проте, для невеликих та «стандартних» проектах, ця функція може тільки заважати;
- е) інтегрована вікі. Допоміжний, але не головний функціонал, який допомагає тримати загальну інформацію у одному місці.
- За допомогою цих критеріїв буде проведено порівняння.

4.3 Порівняння алгоритмів багтрекінгових систем

На Таблицях 2.1.1 та 2.1.2 зображено порівняльну характеристику між розглянутими багтрекінговими системами.

Таблиця 2.1.1 Порівняльна таблиця

Назва	Розширений пошук	Високий рівень захисту	Трекери часу	Email повідомлення	Кастомні WF
Bugzilla	Так	Так	Так	Так	Так
REDMINE	Ні	Ні	Так	Так	Так
MANTIS	Так	Ні	Так	Так	Так
TRAC	Ні	Ні	Так	Так	Ні
Jira	Так	Так	Так	Так	Так
Gant pro	Так	Ні	Так	Так	Ні

Таблиця 2.1.2 Порівняльна таблиця

Назва	Пошук дуплікатів	RSS підтримка	Відстежування у реальному часі	Інтегрова на вікі	Підтримка декількох БД
Bugzilla	Так	Ні	Ні	Так	Так
REDMINE	Так	Ні	Ні	Ні	Так
MANTIS	Ні	Так	Ні	Так	Так
TRAC	Так	Ні	Ні	Так	Ні
Jira	Так	Ні	Так	Так	Так
Gant pro	Ні	Ні	Так	Ні	Так

Усі розглянуті системи можуть бути використані у ході розробки програмного продукту, проте найбільш функціональними є системи Bugzilla та Jira. Проте, не зважаючи на те, що Jira має трішки більший функціонал, у зв'язку з її вартістю, вона може підійти не для всіх типів компаній. Наприклад для невеликих команд або компаній, зайвий функціонал може не лише не використовуватися, а й заважати, відволікаючи від дійсно важливих речей.

При виборі багтрекінгової системи, важливо мати уяву про:

- як багато людей працює у вас в команді, скільки співробітників будуть користуватися інструментом;
- чи є співробітники, що працюють віддалено;
- наскільки рішення масштабоване при зростанні кількості користувачів;
- наскільки гнучка цінова політика компанії, що пропонує послугу;
- як плануєте впроваджувати інструмент і на який рівень підтримки розраховуєте.

Якщо, наприклад, обирати Jira на кожен проект, для усіх компаній, то можливо побачити, що, наприклад, кастомізація WF іноді заважає, оскільки ускладнює роботу команди, та відстеження з боку замовника. Також важливо, наскільки широкий доступ повинен бути, та який існує розподіл ролей та обов'язків. Чи повинен замовник мати можливість переглядати хід роботи та інше. На кожному проекті можуть бути свої критерії, та специфічні вимоги. Тому не має однієї системи, котра підійде на кожний окремий проект.

І в залежності від цих даних, для досягнення максимальної ефективності, потрібно обирати найбільш зручну систему під кожний окремий проект.

Розглянемо приклад, вибору оптимальної системи для проекту.

Наприклад, ми маємо компанію на 200+ людей, котра розробляє глобальний продукт. Частина працівників працює з дому, та компанія розробляє продукт, коли замовник не має можливості відстежувати хід роботи. Також компаній веде набір нових працівників.

Для порівняння, ми будемо враховувати, на скільки важливим є той чи інший параметр для нашого продукту.

Якщо цей функціонал є у системі, 1 *коефіцієнт значимості для проекту, якщо відсутні – (-1) *коефіцієнт.

По-перше, потрібно визначити коефіцієнти для кожного пункту від 1 до 10:

- а) розширений пошук – в рамках одного проекту не є важливою частиною, оскільки немає необхідності шукати по декільком схожим проектам. Проте, пошук по декільком критеріям дозволяє полегшити роботу, тож коефіцієнт = 3;
- б) високий рівень захисту – для великих проектів, дуже важливим компонентом є захист. Тому для нашого прикладу коефіцієнт буде = 10;
- в) трекер часу – в рамках розробки глобального продукту, коли сплановано час та дату, коли необхідно закінчити той чи інший функціонал – еркер часу не є важливим критерієм, тож коефіцієнт = 1;
- г) Email повідомлення – на великому проекті є важливою частиною, бо можеш бути часті зміни, та не такі великі, щоб кожного разу повідомляти всім про це усім. Також важливо, що є співробітники, котрі працюють віддалено, та не мають можливості дізнатися про мінорні зміни особисто, тож коефіцієнт складає 8, оскільки є важливим, та не критичним;
- д) кастомні WF- цей критерій дуже сильно залежить від типу проекту, тож у нашому прикладі коефіцієнт складає 5;
- е) інтегрована вікі – на великому проекті, де можуть бути стандарти, та велика кількість різної інформації, місце, до можливо усе це зберігати стає важливою частиною. Також при наборі нових працівників дозволяє зберегти час на їх навчання. Тож для цього прикладу коефіцієнт буде = 8, оскільки є дуже корисним, та проте не критичним.

На таблиці 2.2 ми бачимо порівняння розглянутих систем на основі коефіцієнтів, котрі ми визначили раніше.

Таблиця 2.2 – порівняння багтрекінгових систем

Назва	Розширений пошук	Високий рівень захисту	Трекери часу	Email повідомлення	Кастомні WF	Вікі & чат	Загалом
Bugzilla	3	10	1	8	5	8	35
REDMINE	-3	-10	1	8	5	8	9
MANTIS	3	-10	1	8	5	8	15
TRAC	-3	-10	1	8	-5	-8	-17
Jira	3	10	1	8	5	8	35
Gant pro	3	-10	1	8	-5	8	5

Як ми бачимо, з таблиці 2.2, найбільш відповідні системи для цього проекту – це Jirata Bugzilla. Та оскільки Jiraє більш складною для освоєння, краще надати перевагу Bugzilla

4.4 Результати порівняння багтрекінгових систем

У результати аналізу та порівняння існуючих багтрекінгових систем не було виявлено однієї, котра могла підійти під усі типи проектів. Тож для кожного

окремого проекту треба обирати систему, котра буде відповідати усім специфікам проекту. На прикладі таблиці 2.2 можливо побачити приклад вибору системи.

По-перше, необхідно визначити вимоги до програмного продукту, на основі яких подалі буде обрана та чи інша система.

По друге, необхідно визначити системи, котрі будуть відповідати мінімальним критеріям програмного продукту, а саме:

- а) чи є специфічні вимоги до програмного продукту;
- б) чи повинен замовник мати доступ до багтрекінгової системи, оскільки веб-додаток дозволяє зробити це зручніше;
- в) яка кількість людей буде працювати з проектом;
- г) яка кількість паралельних проектів є у компанії;
- д) який метод управління використовується на проекті ті інші.

Потім необхідно порівняти системи за обраними критеріями (приклад порівняння таблиця 2.2).

Та проте, загальну перевагу мають багтрекінгові системи Jirata Bugzilla, оскільки мають найбільші функціональні можливості. Та якщо у компанії велика кількість проектів з різними вхідними даними, то доцільніше буде обрати одну з цих систем, так як вони є найбільш унікальними.

ВИСНОВКИ

В результаті атестаційної роботи був проведений аналіз предметної галузі, було проаналізовано існуючі багтрекінгові системи та проведено порівняння на основі якого було виявлено найбільш оптимальний алгоритм вибору системи.

Спочатку було проаналізовано існуючі багтрекінгові системи, для загального розуміння предметної області. Потім було виявлені критерії, котрі підходять для розглянутого проекту, та за цими критеріями було проведено порівняльний аналіз розглянутих багтрекінгових систем.

В процесі дослідження були отримано наступні результати:

- найбільшим набором функціональних можливостей має Jira;
- для досягнення максимального результату треба обирати багтрекінгові системи в залежності від необхідності;
- важливо враховувати усі фактори та специфіки кожного продукту окремо.

У результаті порівняння було отримано алгоритм вибору багтрекінгової системи, використовуючи який, можливо підібрати систему до кожного окремого проекту.

У кожної багтрекінгової системи існує багато плюсів та мінусів, і вибір кожної із них залежить від специфіки проекту.

Це дослідження є важливим для вибору ефективної багтрекінгової системи, яка допоможе у розробці, та тестуванні програмного продукту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. J. Aranda and G. Venolia, The secret life of bugs: Going past the errors and omissions in software repositories: ICSE'09 Proceedings of the 31st International Conference on Software Engineering, 2009.
2. N. Bettenburg, S. Just, A. Schroter, C. Weiss, R. Premraj, and T. Zimmermann, "What makes a good bug report?", In FSE'08: Proceedings of the 16th International Symposium on Foundations of Software Engineering, pages 308–318, November 2008.
3. Zou, F. and Davis, J., "Improving Software Reliability Modelling Using Machine Learning Techniques", International Journal of Software Engineering and Knowledge Engineering, 2008 Vol. 18(7), PP. 965-986
4. S. Just, R. Premraj, and T. Zimmermann, "Towards the next generation of bug tracking systems", In VL/HCC'08 Proceedings of the 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, pages 82–85, September 2008.
5. Black, R. 1999. Managing the Testing Process: The tools you need. Retrieved January 20, 2010 from <http://library.books24x7.com>
6. JIRA Software - Issue Tracking for Software Teams Atlassian <https://www.atlassian.com/software/jira>
7. CATOLINOG.FERRUCCIF.1,PALOMBAF, ZAIDMANA. "NOT ALL BUGS ARE THE SAME: UNDERSTANDING, CHARACTERIZING, AND CLASSIFYING BUG TYPES" In [JOURNAL OF SYSTEMS AND SOFTWARE](#) pp. 165-181, 2019
8. Груздо І.В., Соловійова К.О., Піщухіна О.О., Данилов А.Д. Аналіз, розробка та управління вимогами: Навч. посібник. Харків: ХНУРЕ, 2015. 276с., 2015
9. Singh, V. B., and Krishna Kumar Chaturvedi. "Bug tracking and reliability assessment system (btras)." International Journal of Software Engineering and Its Applications 5.4 (2017): 1-14.

10. Голян Н.В. Безсмертний О. П. Груздо І.В. Використання метрик на різних етапах тестування. Журнал – Переяслав-Хмельницький, 2018. – Вып. 12(44), ч. 1. – С. 6–11.
11. CHEN L., HUANG S., SUN J., HUI Z., YANG S. [INTERNATIONAL JOURNAL OF PERFORMABILITY ENGINEERING](#) «BUG REPORT CLASSIFICATION BASED ON VECTOR SPACE MODEL» pp. 2071-2080, 2019
12. Jalbert, Nicholas, and Westley Weimer. "Automated duplicate detection for bug tracking systems." 2016 IEEE International Conference on Dependable Systems and Networks with FTCS and DCC (DSN). IEEE, 2016.
13. В.В. Бобровська., Голян Н.В «Извлечение требований к разработке мобильных приложений с использованием бизнес-моделей» Радиоэлектроника и молодежь в XXI веке: 16-й Междунар. молодеж. форум, 15–16 грудня 2011 р.: тези доп. – Харків. – 2011. – С. 194–195.
14. M. A. Storey, A. Zagalsky, F. Filho, L. Singer, and D. German. How social and communication channels shape and challenge a participatory culture in software development. IEEE Transactions on Software Engineering, PP(99):1–1, 2016.
15. Dhaval Vyas, Tara Capel, Deven Tank, and David Shepherd. Understanding the use of a bug tracking system in a global software development setup. In Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction, pages 222–226. ACM, 2015.

ДОДАТОК А
СЛАЙДИ ПРЕЗЕНТАЦІЇ

Атестаційна робота магістра

**Дослідження та порівняльний аналіз алгоритмів управління багтрекінговими
системами**

Виконав: ст. гр. ПЗм-18-3

Чернявський М.А.

Науковий керівник:

Доц. Лановий О.Ф.

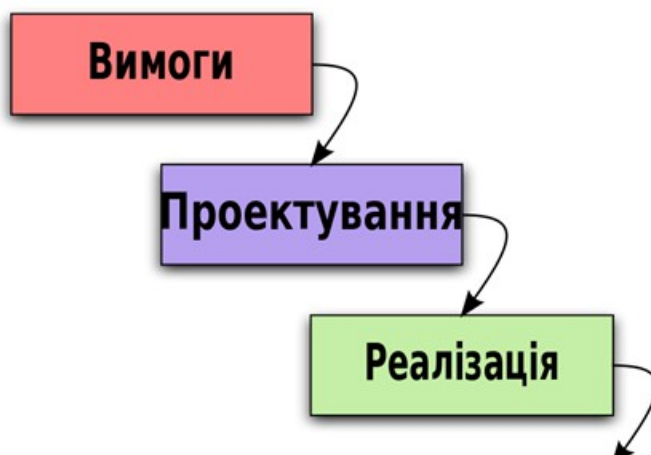
МЕТА РОБОТИ

- Дослідити предметну область
- Визначити критерії для порівняння
- Проаналізувати існуючі багтрекінгові системи
- Провести порівняння
- Виявити алгоритм вибору систем

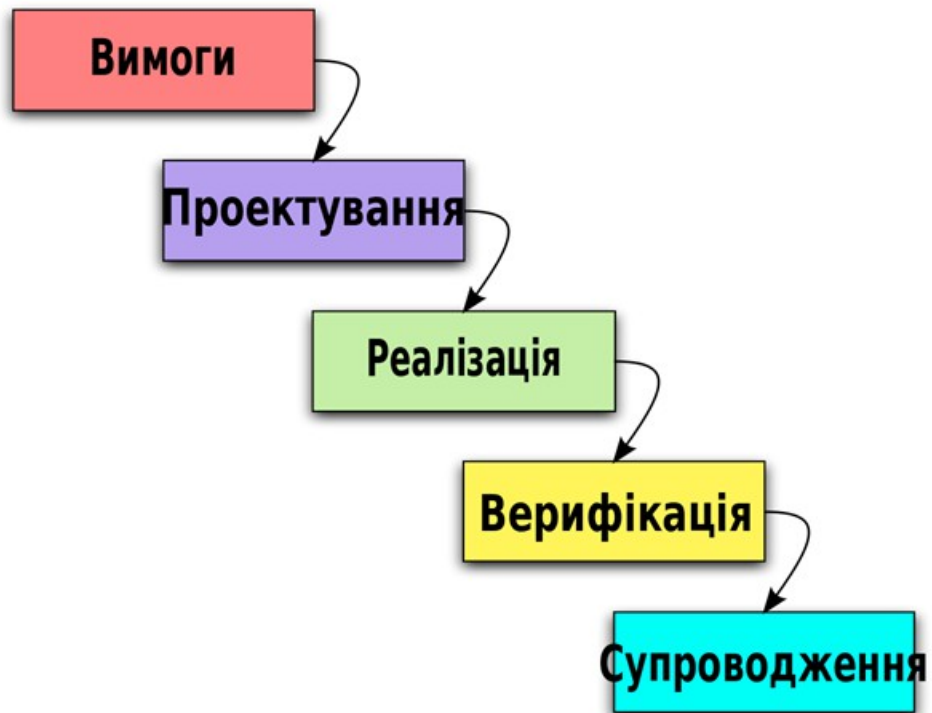
Передумови

Нам всім відома популярна приказка «Людині властиво помилятися». Основна мета розробників програмного забезпечення - дізнатися, яка помилка була допущена, виправити її і потім витягти уроки, щоб не допускати подібних помилок надалі. Важливо, щоб інформація у звітах була детальна та актуальна

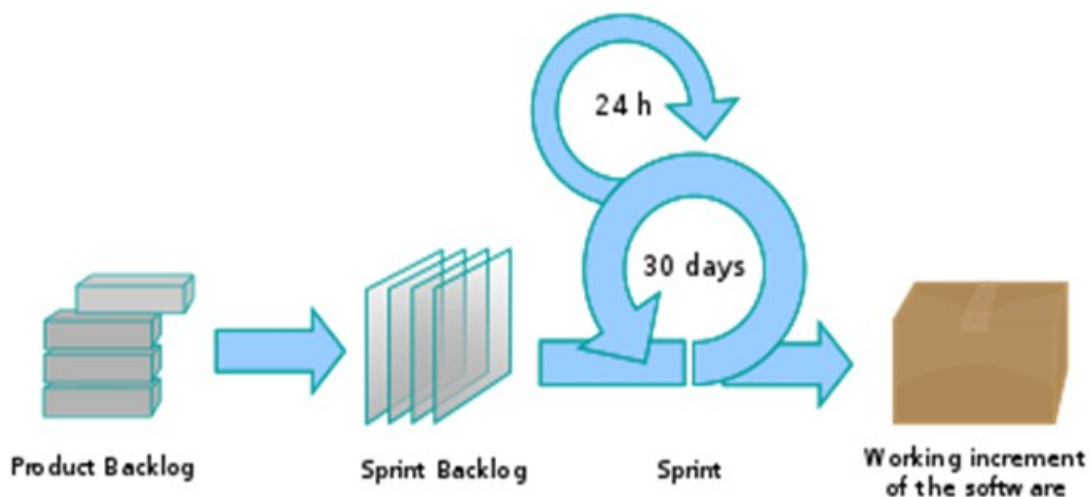
Традиційна модель ЖЦ



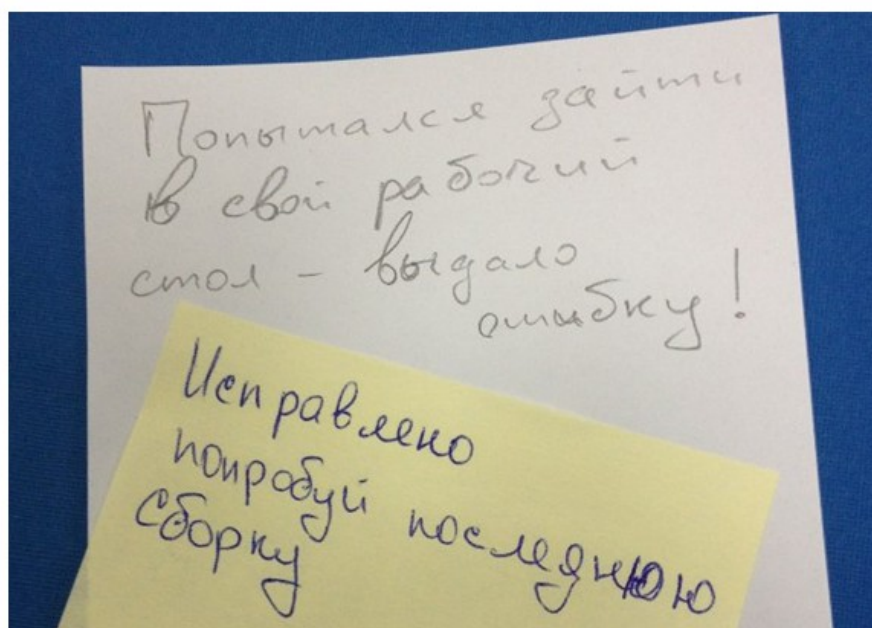
Традиційна модель ЖЦ



ЖЦ Scrum



Простий баг-репорт



BUG

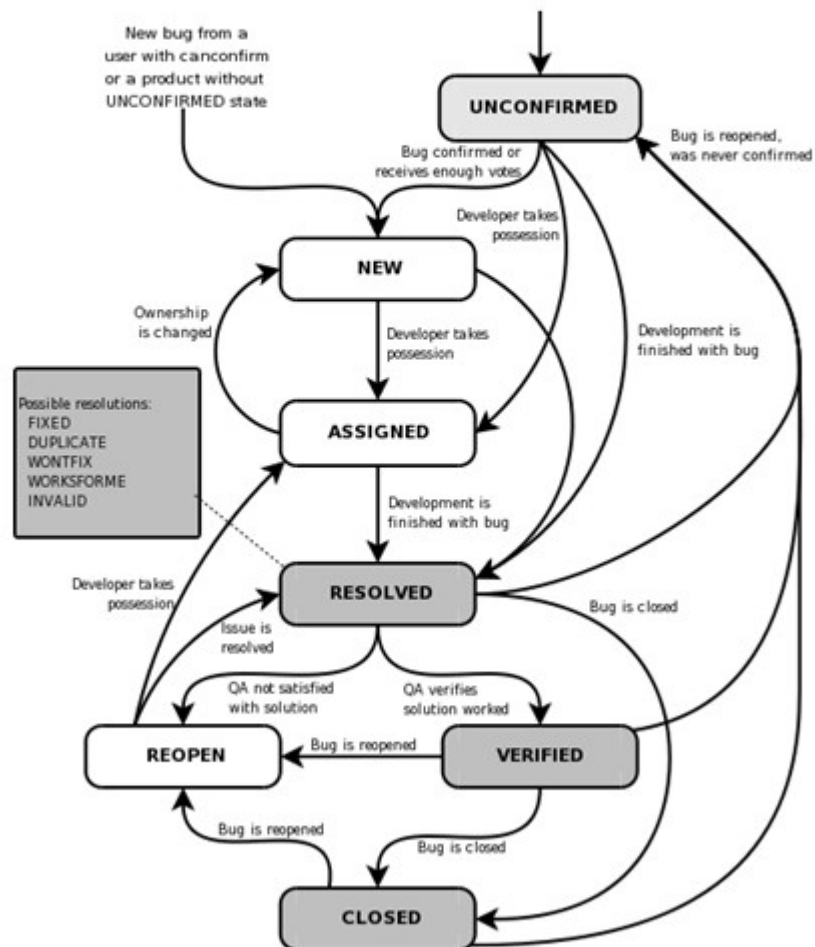
Програмна помилка (баг) – означає помилку в програмі або в системі, через яку програма видає несподівану поведінку і, як наслідок, результат. Більшість програмних помилок виникають через помилки, допущені розробниками програми в їх вихідному коді, або в її дизайні.

Bugzilla

- Bugzilla в даний час є однією з найпопулярніших систем для відстеження дефектів, що охоплюють більшість функцій, очікуваних від ефективної системи стеження за вадами.
- Bugzilla підійде для великих проектів, оскільки має багатий функціонал. Та для не великих проектів може бути зайвим більша частина функціоналу, та відволікати від дійсно важливих речей.

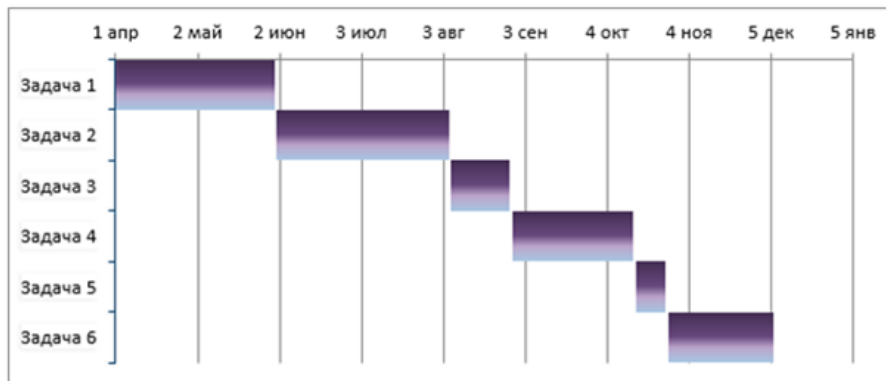


ЖЦ Bugzilla



REDMINE

- Redmine – система, котра підійде більш для невеликих проектів, або до проектів з чіткими вимогами, тобто проекти, що розробляються за каскадною моделлю.



MANTIS

- Mantis – це безкоштовне, доступне для скачування програмне забезпечення для відстеження помилок з відкритим вихідним кодом.
- Система має гнучкі можливості конфігурації, що дозволяє налаштовувати її не тільки для роботи над програмними продуктами, а й в якості системи обліку заявок для Helpdesk
- Система є веб-додатком, тому не вимагає для

Trac

Функція тимчасової шкали відображає всі поточні і події попередніх проектів полегшували відстеження дій проекту. Використовується на проектах, де можуть бути часті зміни.

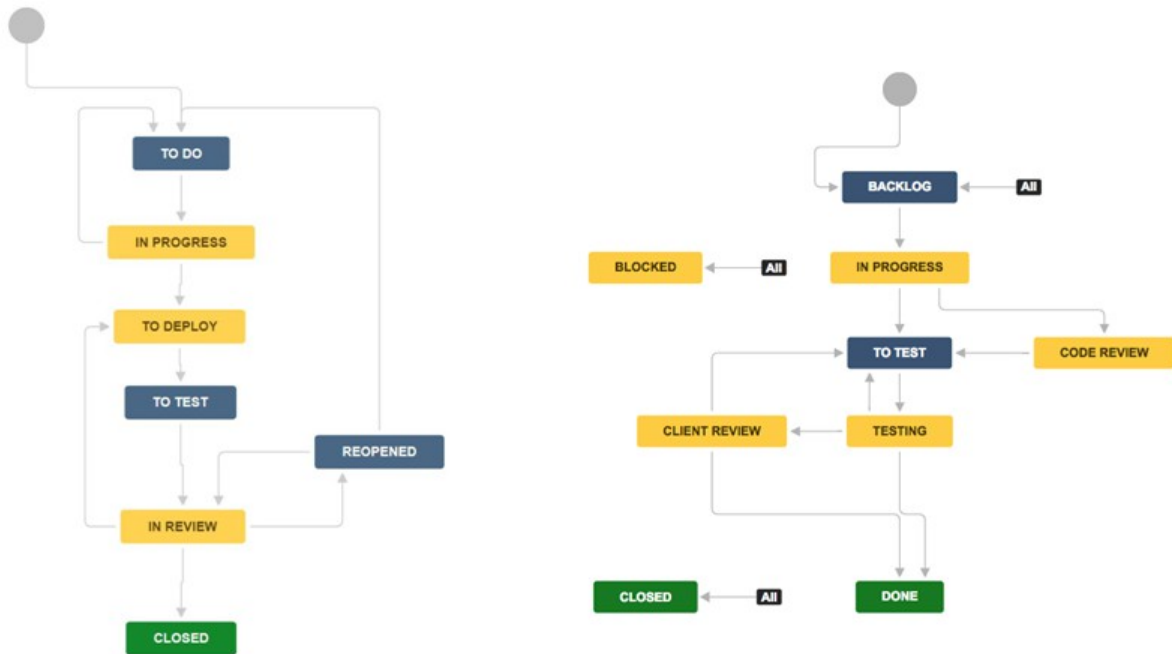


Jira

- це web-базуватися засіб для управління проектами та завданнями. JIRA може застосовуватися у всіх випадках, коли необхідно організувати роботу співробітників, ефективно призначати їм завдання, мати миттєві засоби контролю виконання

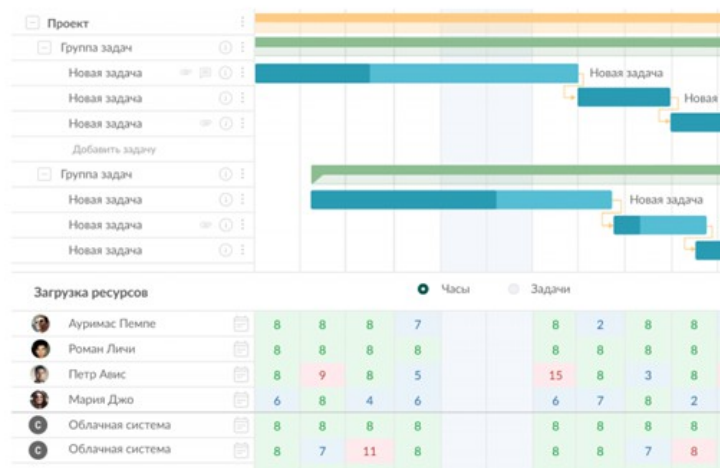


Модіфікація WF



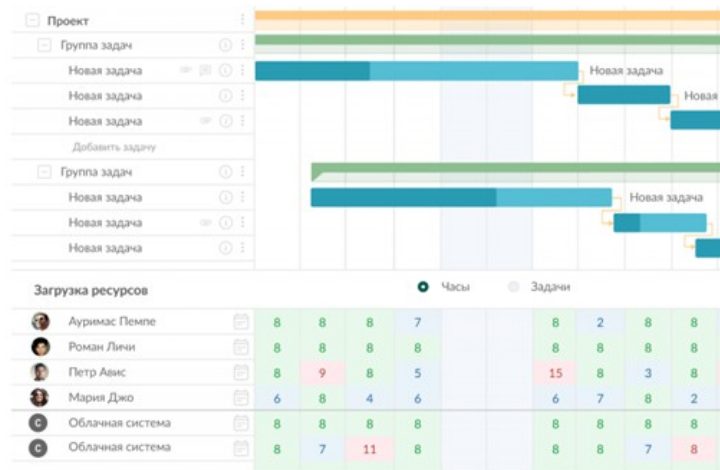
Gantt PRO

Це доступний, та зручний інструмент для роботи у невеликих командах, та зручний для відстежування зайнятості кожного окремого працівника



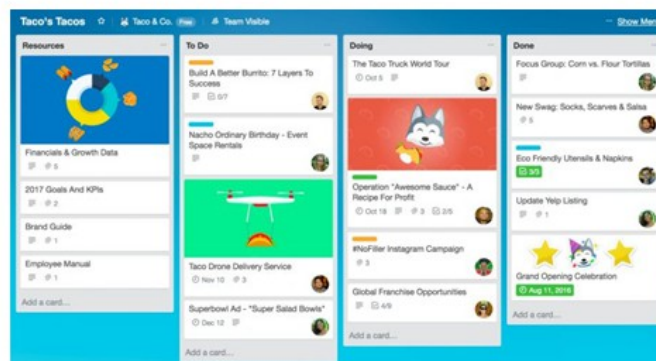
Gantt PRO

Це доступний, та зручний інструмент для роботи у невеликих командах, та зручний для відстежування зайнятості кожного окремого працівника



Trello

Попрете, що Trello не є багтрекінговою системою її теж варто розглянути, оскільки невеликі компанії та команди використовують цей програмний продукт, для відстежування етапів проекту



Критерії класифікації

- Розширений пошук;
- Високий рівень захисту;
- Трекер часу;
- Кастомні WF;
- Інтегрована вікі;
- Email повідомлення.

Порівняльна таблиця

Назва	Розширений пошук	Високий рівень захисту	Трекери часу	Email повідомлення	Кастомні WF	інтегрована вікі & Чат	Загалом
Bugzilla	3	10	1	8	5	8	35
REDMINE	-3	-10	1	8	5	8	9
MANTIS	3	-10	1	8	5	8	15
TRAC	-3	-10	1	8	-5	-8	-17
Jira	3	10	1	8	5	8	35
Gant pro	3	-10	1	8	-5	8	5

Загальну перевагу мають багтрекінгові системи Jira та Bugzilla, оскільки мають найбільші функціональні можливості.

Дякую за увагу!