

МЕТОДОЛОГІЯ KANBAN ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Білоус М.Ю., Іщенко М.Д.

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки 14

E-mail: mykhailo.ishchenko@nure.ua, maryna.bilous@nure.ua

Анотація: У даній статті проведено аналіз актуальності Agile принципів та детально розглянута методологія Kanban. Визначені основні цінності і принципи Agile, їх переваги та недоліки. Наведені переваги і недоліки Kanban методології.

Ключові слова: Agile, Kanban, система, керування проектами, програмне забезпечення.

KANBAN METHODOLOGY FOR SOFTWARE DEVELOPMENT

Bilous M., Ishchenko M.

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky av, 14

E-mail: mykhailo.ishchenko@nure.ua, maryna.bilous@nure.ua

Abstract: This article analyzes the relevance of Agile principles and examines the Kanban methodology in detail. The main values and principles of Agile, their advantages and disadvantages are defined. The advantages and disadvantages of the Kanban methodology are given.

Key words: agile, kanban, system, project management, software.

Розробка програмного забезпечення (ПЗ) – це процес створення проєкта, який буде працездатний, якісний і надійний. ПЗ може бути з галузі Web або робототехніки, це може бути, взагалі інформаційно-пошукова система або система для авторизації розрахунків [1 – 6]. Кожне програмне забезпечення розроблюється на базі заздалегідь розробленої концепції, моделі [1 – 6] та плану, хоча у сучасної економіки, планування іноді втрачає свою актуальність. Це обумовлено умовами, що постійно змінюються, а також низькою мобільністю компаній при використанні планування. Виникає необхідність у системі, що дозволяє підвищити адаптацію бізнесу до змінних умов та різних факторів. Як така система може виступати agile-методологія.

Більшість сучасних менеджерів вважають, що Agile підхід є новим у практиці менеджменту. Однак свій початок Agile бере в 1970 році, коду Уїнстон Райс представив світу методику «управління розробкою великих програмних систем». У 90-х роках було розроблено методології Scrum, Kanban, які безпосередньо відносяться до гнучких методологій. У 2001 році був опублікований Agile Manifesto, який містить у собі 4 базові цінності та 12 принципів [7 – 12].

Виділимо основні цінності:

- люди та взаємодія важливіші за процеси та інструменти;
- працюючий продукт важливіший за вичерпну документацію;
- співробітництво із замовником важливіше за погодження умов контракту;
- готовність до змін важливіша за дотримання початкового плану.

Основні принципи:

- задоволення потреб клієнта за рахунок раннього та безперебійного постачання цінності;
- зміни вимог можуть відбуватися навіть наприкінці розробки;
- видаються часті проміжні результати;
- тісне, щоденне спілкування замовника з розробниками протягом усього проєкту;
- проєктом займаються мотивовані особи, які забезпечені необхідними умовами для роботи, підтримкою та довірою;

- рекомендований метод передачі – особиста розмова (віч-на-віч);
- працюючий продукт чи процес – найкращий вимірник прогресу;
- спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений термін;
- постійна увага покращення майстерності та зручного дизайну;
- простота – мистецтво не робити зайвої роботи;
- найкращі технічні вимоги, дизайн та архітектура виходять у самоорганізованій команді;
- постійна адаптація до обставин, що змінюються. Команда повинна систематично аналізувати можливі способи покращення ефективності та відповідно коригувати стиль своєї роботи.

Виходячи з основних принципів та цінностей, можна зробити висновок, що agile – це швидкий та гнучкий підхід до управління проектами на основі принципів співробітництва, адаптивності та безперервного вдосконалення, а також сегментації основного проекту на дрібніші структурні одиниці.

Можна виділити такі переваги гнучкої системи проектного управління:

- проект прозорий та зрозумілий для замовника на всіх етапах;
- забезпечує швидке коригування курсу на основі зворотного зв'язку із заінтересованими сторонами;
- надається свобода дій для команди проекту, що дозволяє підвищити ефективність та стимулювати творчий підхід;
- усі члени команди тісно взаємодіють між один одним.

Але, як і будь-який метод, він має свої недоліки:

- постійна зайнятість команди;
- не застосовується для проектів з чіткими вимогами та обсягами;
- нечіткі терміни та обсяги витрат спочатку можуть насторожити замовника;
- вимагає постійного коригування документації та завдань для команди проекту.

Розглянемо детальніше методологію Kanban для розробки програмного забезпечення.

Основним методом візуалізації в Канбані є дошка Канбан (рис. 1). У виробничому середовищі це фізична дошка, де розміщуються квитки щодо попиту на ресурси. Хоча дошки можуть використовуватися розробниками програмного забезпечення, віртуальні дошки Kanban, очевидно, є кращим вибором для команд, які часто географічно розосереджені та працюють у віртуальному середовищі.

Kanban – це метод, який значною мірою підтримує гнучкі команди, тому інструменти гнучкого керування проектами та ALM часто надають дошку Kanban. Цю дошку Kanban можна використовувати для відстеження завдань і їх статусів, надаючи всім учасникам простий огляд того, як просувається робота в порівнянні з планами [10 – 12].



Рисунок 1 – Дошка Kanban

Загальний макет дошки Kanban, що використовується при розробці програмного забезпечення, містить перелік усіх завдань, які потрібно виконати, які виконуються, які потрібно перевірити та які виконуються, в окремих стовпцях.

Узагальнений макет дошки містить такі стовпці:

- «зробити» (to-do);
- «в процесі» (in progress);
- «зроблено» (done).

Але також дошки можуть містити стовпці: обговорюється (backlog), узгоджується (ready), код пишеться (coding), тестується (testing), затверджується (approval) і робиться (done)

Обмеження незавершеної роботи (WIP) можуть бути встановлені таким чином, щоб розробники завжди мали повне завдання, але не були перевантажені завданнями. Учасники команди будуть переміщувати картки з одного стовпця в інший під час роботи.

Kanban – це простий та ефективний метод відстеження потоку роботи та виявлення вузьких місць у процесі розробки.

Виділимо переваги Kanban в розробці програмного забезпечення.

- обмеження незавершеної роботи (WIP): щоб підвищити ефективність, слід обмежити кількість елементів у статях незавершеної роботи. Це допомагає гарантувати, що нова робота «затягується» у виробництво лише тоді, коли це необхідно, тобто коли є доступна потужність у межах локального ліміту WIP;

- простіші зміни: картонні назви стовпців представляють різні стани робочого процесу. Візуалізація потоку роботи є ключовою для розуміння того, як відбувається робота, і забезпечує легше прийняття рішень щодо змін;

- допомагає керувати робочим процесом: слід відстежувати, вимірювати та звітувати про перебіг роботи в кожному стані процесу. Таким чином, дошка Kanban також корисна для керівників проектів, оскільки вона дає їм уявлення про продуктивність команди, а також може допомогти їм виявити закономірності, що обмежують продуктивність;

- підтримує вирішення проблем і управління ризиками: завдяки чіткому розумінню того, як все працює та як насправді виконується робота, можна перейти до більш раціонального, об'єктивного обговорення питань, проблем і ризиків. Використання цього методу заохочує членів команди висловлювати пропозиції щодо покращення;

- сприяє співпраці та експерименту: Kanban заохочує невеликі, безперервні, поступові та еволюційні зміни, які залишаються незмінними. Наприклад, Kanban може ефективно підтримувати щоденні стендап-зустрічі Scrum, оскільки він надає швидкий огляд завдань, статусів і WIP для кращого розподілу ресурсів. Це допомагає впроваджувати безперервні, поступові та еволюційні зміни, а також підвищує мотивацію та самоврядність команд.

Виділимо недоліки Kanban:

- вимагає постійної наявності завдань, як на конвеєрі. Якщо в якийсь момент вони закінчуються, почнеться процес руйнування. Тож, якщо проєкт має періоди нерівномірної роботи, впровадити Kanban буде важко;

- може миттєво загальмувати. Щоб усе чітко працювало, ви повинні мати добре налаштовані процеси, і кожен у команді має знати, що входить у його роботу. Якщо у вас цього немає, Kanban швидко розпадеться. Якщо співробітники не впевнені в тому, що відноситься до їхньої роботи та зони відповідальності, завдання можуть «провисати», а це спричинить ефект доміно для інших членів команди;

- немає часових обмежень. В Kanban немає строків, для виконання завдання дається стільки часу, скільки потрібно. Це може створити проблеми з дедлайнами проєкту. Для боротьби з цим команда може визначити час на роботу з кожною картою;

- можна «втратити» задачі з високим пріоритетом. Оскільки співробітники самі беруть завдання в роботу, вони можуть обрати ті, які мають низький пріоритет для бізнесу. Отже, всі задачі на дошці мають бути ретельно перевірені та визначені пріоритетними.

В результаті проведеного огляду «Agile» методологій, більш детально розглянуто саме Kanban, бо вона є однією з найпопулярніших як для невеликих, так і для малих проектів завдяки орієнтованості на задачі, а саме: на зменшення кількості одночасно виконуваних задач і зменшення часу виконання кожної задачі, гнучкість у прийнятті рішень і частоту змін пріоритетів, тісну взаємодію між усіма членами команди. Наукова цінність наведеного методу полягає у підвищенні ефективності та покращенні роботи наявних методів організації розробки програмного забезпечення. У ході подальших досліджень планується вдосконалення існуючої моделі з метою підвищення ефективності системи.

ЛІТЕРАТУРА

1. Lyashenko, V. Recognition of Voice Commands Based on Neural Network / V. Lyashenko, F. Laariedh, S. Sotnik, M. Ayaz Ahmad // TEM Journal. – 2021. – Volume 10. – Issue 2. – P. 583-591.
2. Sotnik, S. Developing the information search system for selecting the moulds forming elements / S. Sotnik, V. Nevliudova, I. Malaya // Innovative technologies and scientific solutions for industries (ITSSI). – 2(2). – 2017. – P. 86–92.
3. Sotnik, S. Key Directions for Development of Modern Expert Systems / S. Sotnik, Z. Deineko, V. Lyashenko // International Journal of Engineering and Information Systems (IJEAIS) – Vol. 6 Issue 5. – 2022. – P. 4 – 10.
4. Sotnik, S. Features of Database Types / Z. Deineko, S. Sotnik, O. Vovk, V. Lyashenko // International Journal of Engineering and Information Systems (IJEAIS). – 2021. – Vol. 5 (10). – P. 73 – 80.
5. Sotnik S. Developing the information search system for selecting the moulds forming elements / S. Sotnik, V. Nevliudova, I. Malaya // Innovative technologies and scientific solutions for industries (ITSSI). – 2017. – P. 86 – 92.
6. Sotnik, S. Information model of plastic products formation process duration by injection molding method / M.H. Al-Sherrawi, , A.M. Saadoon, S. Sotnik, V. Lyashenko // International Journal of Mechanical Engineering and Technology. – 2018. – № 9(3). – P. 357 – 366.
7. Abu-Jassar, A.T. Some Features of Classifiers Implementation for Object Recognition in Specialized Computer systems / A.T. Abu-Jassar, Y.M. Al-Sharo, V. Lyashenko, S. Sotnik // TEM Journal [this link is disabled](#). – 2021. – Vol. 10. – Issue 4. – P. 1645 – 1654.
8. Corona, E. A Review of Lean-Kanban Approaches in the Software Development / E. Corona, F.E. Pani // Wseas transactions on information science and applications. – 2013. – P. 1 – 13.
9. Reddy, A. The Scrumban [R] Evolution: Getting the Most Out of Agile, Scrum, and Lean Kanban / A. Reddy – English: AddisonWesley Professional, 2015. – 384 p.
10. Kirichek, G. Decentralized System for Run Services / G. Kirichek, V. Tymoshenko, O. Rudkovskyi, S. Hrushko // In: CEUR Workshop Proceedings 2353. – 2019. – P. 860 – 872.
11. Kirichek, G. Implementation of web system optimization method / G. Kirichek, S. Skrupsky, M. Tiahunova, A. Timenko // In: CEUR Workshop Proceedings 2608. – 2020. – P. 199 – 210.
12. Ahmad, M.O. Kanban in software development: A systematic literature review / M.O. Ahmad, J. Markkula, M. Oivo // In Software Engineering and Advanced Applications (SEAA). – 2013. – P. 1 – 9.