**Міністерство освіти і науки України**

**Національний університет «Запорізька політехніка»**

кафедра програмних засобів

**РЕФЕРАТ**

з дисципліни "Верифікація цифрових систем"

на тему:

"Методологія розробки KANBAN"

Виконав:

студент групи КНТ-227 Ю. Є. Горіченко

Прийняв:

к.т.н., професор Г. В. Табунщик

2019

**ЗМІСТ**

[ВСТУП 3](#_Toc22671955)

[Історія виникнення 4](#_Toc22671956)

[Принципи Kanban 5](#_Toc22671957)

[Ключові практики методу Kanban 5](#_Toc22671958)

[Використання Kanban в IT 7](#_Toc22671959)

[Програми для Kanban 10](#_Toc22671960)

[Переваги та недоліки Kanban 11](#_Toc22671961)

[ВИСНОВКИ 12](#_Toc22671962)

[ЛІТЕРАТУРА 13](#_Toc22671963)

# **ВСТУП**

Для розробки програмного забезпечення у невеликих компаніях зручно використовувати гнучку методологію розробки. Це маніфест, що містить основні цінності та принципи, на яких базуються підходи до управління проектами, який вирішує проблеми традиційного проектного менеджменту. Він підходить для інноваційних проектів. Набагато менше він підходить для процесної діяльності. Ці підходи мають на увазі інтерактивну розробку, з періодичними оновленнями від вимог замовника і їх реалізацію за допомогою самоорганізованих команд, сформованих з експертів різного профілю. Під терміном «гнучка методологія розробки» слід розуміти підходи на основі даного маніфесту, або фреймворки. Існує безліч фреймворків, підходи яких базуються на Agile, наприклад: Scrum, Kanban, Extreme programming, FDD, DSDM та ін. Суть в тому, що робота із застосуванням гнучкої методології складається з серії коротких циклів (ітерацій), тривалістю 2-3 тижні. Кожна ітерація включає в себе етапи планування, аналізу вимог, проектування, розробку, тестування і документування. По завершенню кожної ітерації команда пред’являє замовнику результати роботи, наприклад, первинну версію продукту або частину функціоналу, яку можна подивитись, оцінити, протестувати, а потім доопрацювати або скорегувати. Таким чином, замовник контролює розробку і може на неї відразу впливати. Після кожного етапу, на основі виконаної роботи, команда підводить підсумки і збирає нові вимоги, на підставі чого вносить корективи у план розробки продукту.

# **Історія виникнення**

Kanban – японський термін, який почали використовувати для виробництва в 60-х роках 20-го століття в компанії Toyota. В основу цього принципу покладено конвеєрний метод виробництва, а також різні швидкості виконання окремих технологічних операцій на виробництві. Це означає, що при будь-якому виробництві є основне виробництво («головний конвеєр») і додаткове виробництво («додаткові конвеєри»). Темп випуску кінцевих виробів задає головний конвеєр, в той час як додаткові конвеєри не можуть прискорити темп випуску виробу, але можуть сповільнити його, у разі несвоєчасного випуску необхідних деталей.

Термін Kanban має дослівний переклад: «Kan» означає видимий, візуальний, і «ban» означає картка або дошка. На заводах Toyota картки Kanban використовуються для того, щоб не заставляти склади і робочі місця заздалегідь створеними запчастинами. Ці картки кріпилися до тари з деталями. На таких бірках вказувалася інформація про номер і кількість деталей, який відділ їх відправляє і куди вони повинні прибути. Працівник, який безпосередньо займався монтажем і складанням машин – нижній потік – забирав деталі з тари, на якій був прикріплений «канбан» з запитом для складу. Картка знімалася і разом з порожнім ящиком передавалася транспортувальників на склад. Там інший працівник вже підготував нову тару з запчастинами, на якій кріпився виробничий «канбан» – бірка з інформацією про вироблені запчастини. Виробничий «канбан» замінювався на «канбан» із запитом для складу і відправлявся на виробничу лінію запчастин – верхній потік. Тому створювалося саме ту кількість деталей, яке вказувалося в картці. Тара з новими запчастинами ставилася транспортувальників на монтажну лінію.

# **Принципи Kanban**

Kanban заснований на чотирьох основних принципах:

1. Почніть з того, що ви маєте зараз. Метод Kanban не описує конкретний набір ролей чи кроків процесу. Він стартує з ролями і процесами, що є у вас зараз, і стимулює постійні інкрементні та еволюційні зміни в системі. Kanban — це метод управління змінами.
2. Погодьтесь домагатись інкрементних, еволюційних змін. Організація (чи команда) повинна погодитись, що постійні, інкрементні зміни — це спосіб покращити систему, і зробити так, щоб ці покращення прижились. Глобальні зміни можуть виглядати більш ефективними, але мають більший ризик провалу, через опір та страх змін в організації. Kanban заохочує постійні невеликі зміни до поточної системи.
3. Поважайте поточний процес, ролі, відповідальності та посади. Дуже ймовірно, що організація має деякі елементи, що працюють задовільно, і їх варто зберегти. Kanban намагається уникнути страхів, домовляючись поважати поточні ролі, відповідальності та посади з метою отримати ширшу підтримку.
4. Лідерство на всіх рівнях. Лідерські дії на всіх рівнях – від окремих працівників і аж до старшого менеджменту — схвалюються.

# **Ключові практики методу Kanban**

Існує шість практичних прийомів, які використовуються для поліпшення роботи у команді і підвищення якості сервісу:

1. Візуалізуйте. Візуалізація процесів роботи допомагає в правильному розумінні змін, що плануються і допомагає впроваджувати їх згідно з планом. Типовим способом візуалізувати процес роботи є використання дошки з колонками і картками. Колонки на дошці позначають різні кроки процесу роботи.
2. Обмежуйте задачі в процесі виконання. Обмеження задач в процесі виконання має на увазі те, що використовується система «витягування» на частинах, або всьому процесі роботи. Система «витягування» працює як один з головних стимулів до постійних покращень в системі. Система «витягування» може бути реалізована, як система Kanban, CONWIP, DBR чи якийсь інший варіант. Критичним елементом є те, що робота, котра перебуває в стані виконання на кожному кроці робочого процесу, є обмеженою, і що нова робота «витягується» в кожен крок, коли з’являється місце в колонці кроку.
3. Керуйте потоком. Кожен перехід між станами в потоці моніториться, вимірюється і звітується. Активне управління потоком дозволяє оцінити позитивні та негативні ефекти змін у системі.
4. Зробіть політики явними. Поки механізм чи процес не стане явним, часто важко чи неможливо здійснювати обговорення щодо його вдосконалення. Без явного розуміння, як все працює, будь-які обговорення проблем стають емоційними та суб’єктивними. З явним розумінням можливо перейти до більш раціональних, емпіричних та об’єктивних обговорень проблем.
5. Створіть цикли зворотного зв’язку. Організації що не створили другий рівень зворотного зв’язку — перегляд операцій, — зазвичай не бачать вдосконалення процесу поза локалізованим рівнем команди.
6. Вдосконалюйте співпрацюючи, розвивайтесь експериментально (використовуючи моделі та науковий метод). Метод Kanban пропагує малі поступові, постійні та еволюційні зміни які приживаються. Коли команди мають спільне розуміння теорій про роботу, процес, ризики, вони більш ймовірно будуть здатними виробити спільне розуміння проблем та запропонувати вдосконалення які будуть результатом консенсусу. Метод Kanban радить використовувати науковий підхід до втілення змін.

# **Використання Kanban в IT**

На сьогоднішній день Kanban має велику популярність у виробництві програмного забезпечення. Чистий Kanban погано працює для продуктових команд, але відмінно працює з командами підтримки, такими як:

* групи підтримки програмного забезпечення, де не важливий «план», але важлива швидкість реагування на зміни;
* групи тестування, що працюють окремо від груп розробки;
* служби підтримки;
* інші приклади «неосновних виробництв».

Окремо необхідно зазначити, що Kanban добре працює в стартапах, де не мають чіткого плану, але активно працюють над розробкою.

При розробці програмного забезпечення команда використовує Kanban-дошку (рис. 1.1).

*Expedite* →

*Task*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Goals* | *Story Queue* | *Elaboration acceptance* | *Development* | *Test* | *Deployment* | *Done!* |
| *Task*  *Task*  *Task* | *Task*  *Task* | *Task* | *Task*  *Task*  *Task* | *Task*  *Task* | *Task* | *Task*  *Task*  *Task*  *Task*  *Task*  *Task* |

*5 3 5 3 1*

Рисунок 1.1 – Приклад Kanban-дошки

Kanban-дошка складається з декількох стовпців:

* «Цілі проекту». Необов’язковий, але корисний стовпець. Сюди можна помістити високоприорітетні цілі проекту, щоб команда їх бачила і всі про них знали. Наприклад, «Збільшити швидкість роботи на 20%» або «Додати підтримку Windows XP».
* «Черга завдань». Тут зберігаються завдання, які готові до того, щоб почати їх виконувати. Завжди для виконання береться верхнє, найбільш пріоритетне завдання і її картка переміщається у наступний стовпець.
* «Опрацювання дизайну». Цей і інші стовпці до «Закінчено» можуть змінюватися, тому що саме команда вирішує, які кроки проходить завдання до стану «Закінчено». Наприклад, в цьому стовпці можуть знаходитися завдання, для яких дизайн коду або інтерфейсу ще не зрозумілий і обговорюється. Коли обговорення закінчені, завдання пересувається в наступний стовпець.
* «Розробка». Тут завдання висить до тих пір, поки його розробка не завершена. Після завершення воно пересувається в наступний стовпець. Або, якщо, архітектура не вірна або не точна – завдання можна повернути в попередній стовпець.
* «Тестування». В цьому стовпці завдання знаходиться, поки воно тестується. Якщо знайдені помилки – повертається в розробку. Якщо ні – пересувається далі.
* «Деплоймент». У всіх проектів свій деплоймент. У когось це значить викласти нову версію продукту на сервер, а у когось – просто зробити коміт коду в репозиторій.
* «Закінчено». Сюди стікер потрапляє тільки тоді, коли всі роботи по завданню закінчені повністю.

У будь-якій роботі трапляються термінові завдання. Заплановані чи ні, але такі, які треба зробити негайно. Для таких можна виділити спеціальне місце (на рис. 1.1 відзначено, як «Expedite»). У «Expedite» можна помістити одну термінову задачу і команда повинна почати її виконувати негайно і завершити якомога швидше. Але може бути тільки одна така задача. Якщо з’являється ще одна — вона повинна бути додана в «Чергу завдань».

Під кожним стовпцем є цифри. Це число завдань, які можуть бути одночасно в цих стовпцях. Цифри підбираються експериментально, але вважається, що вони повинні залежати від числа розробників в команді. Наприклад, якщо ви маєте 8 програмістів в команді, то в рядок «Розробка» ви можете помістити цифру 4. Це означає, що одночасно програмісти будуть робити не більше 4-х завдань, а значить у них буде багато причин для спілкування та обміну досвідом. Якщо ви поставите туди цифру 2, то 8 програмістів, що займаються двома завданнями, можуть втрачати занадто багато часу на обговорення. Якщо поставити 8, то кожен буде займатися своєю завданням і деякі завдання будуть затримуватися на дошці надовго, адже головне завдання Kanban – це зменшення часу проходження завдання від початку до стадії готовності.

Таким чином, Kanban-дошка є дуже корисною та відображає багато корисної інформації.

По-перше, зменшення числа паралельно виконуваних завдань сильно зменшує час виконання кожної окремої задачі. Немає потреби перемикати контекст між завданнями, відстежувати різні сутності, планувати їх і т.д. – робиться тільки те, що потрібно і детальне опрацювання завдання починається тільки тоді, коли завдання починає виконуватися.

По-друге, відразу видно затримки. Наприклад, якщо тестери не справляються з тестуванням, то вони дуже скоро заповнять весь свій стовпець і програмісти, які закінчили нове завдання, вже не зможуть перемістити його в стовпець тестування, тому що він заповнений. Тому, програмісти можуть допомогти тестерам завершити одне із завдань тестування і тільки тоді пересунути нове завдання на місце, що звільнилося. Це дозволить виконати обидва завдання швидше.

По-третє, можна обчислити час на виконання усередненої задачі. Ми можемо позначати на картці дату, коли вона потрапила в чергу завдань, потім дату, коли її взяли в роботу і дату, коли її завершили. За цими трьома точками для хоча б десяти завдань можна вже порахувати середній час очікування в чергу завдань і середній час виконання завдання. А з цих цифр менеджер може вже розраховувати все, що йому завгодно.

Також, потрібно зазначити про важливість двох ролей у Kanban. Це менеджер сервісу поставки (Service Delivery Manager) і менеджер сервісу запитів (Service Request Manager). Перший відповідає за усунення перешкод в потоці поставки. Другий – за управління потоком запитів до сервісу від замовників. Дуже важливо, щоб ці дві ролі були партнерами і працювали в парі.

Отже, Kanban — це «витягуюча» система. В ній створюється баланс між постійним потоком, який усуває витрати на очікування, і мінімальною кількістю роботи в процесі, що знижує ризики перевиробництва. Робота в процесі регулюється за допомогою карток: їх кількість зафіксована, а інструкції в них направляють виконавців нижнього потоку.

# **Програми для Kanban**

Існує багато програм використання Kanban-методу:

* Trello – популярна система Kanban для управління завданнями. Відрізняється візуальною привабливістю і зручним інтерфейсом.
* Kanbantool – безкоштовна дошка для двох користувачів. Підтримка API і touch-інтерфейсів.
* Lean Kit Kanban – безкоштовна дошка для п’яти користувачів з хорошою реалізацією спільної роботи. Підтримка API і можливість імпорту/експорту, широка статистика.
* Kanbanize – повністю безкоштовний сервіс з гідною функціональністю.
* Worksection – український SaaS-додаток для швидкого відстеження та управління проектами. Зараз у функціях крім обліку фінансів, термінів і діаграми Ганта є настроювана Kanban -дошка.

# **Переваги та недоліки Kanban**

Kanban має такі переваги:

* Гнучкість планування. Команда концентрується тільки на поточній роботі, пріоритет завдання виставляється менеджером.
* Високе залучення команди в процес розробки. Завдяки постійним зібранням, прозорості процесів та можливостям самоорганізації працівники сплачуються та виявляють щирий інтерес.
* Менша тривалість циклу. Якщо кілька людей володіє схожими навичками, тривалість скорочується, якщо ж тільки один – з’являється вузьке місце. Тому співробітники повинні ділитися знаннями і тим самим оптимізувати тривалість циклу. Тоді вся команда зможе взятися за роботу, яка зупинилася, і відновити плавний потік.
* Менше вузьких місць. Ліміти роботи в процесі дозволяють швидко знаходити вузькі і проблемні місця, які з'явилися через дефіцит уваги, людей або навичок.
* Наочність. Коли всі виконавці мають доступ до даних, то вузькі місця легше помітити. Kanban-команди, крім самих карток, зазвичай використовують два загальних звіту: графіки управління і сукупного потоку.

Також, Kanban має наступні недоліки:

* Система погано працює з командами чисельністю понад 5 осіб.
* Він не призначений для довгострокового планування.

# **ВИСНОВКИ**

Kanban – це практика, яка допомагає досягти успіху, при цьому використання тільки гнучких методів необов'язково. Важливі зміни досягаються завдяки виключенню втрат часу, управління вузькими місцями і зниження варіативності. Мета цього методу – виробляти тільки необхідну продукцію в необхідній кількості і в потрібний час. Головна перевага системи Kanban полягає в запобіганні перевиробництва.

Kanban добре працює, коли є неясні вимоги навіть на початку; дуже часто виникають несподівані завдання; бажання відкласти ухвалення рішень; проблема не в розробці, а в інших частинах процесу; вузька спеціалізація людей у команді; підходить для малих і середніх компаній, для підрозділів і проектів.

# **ЛІТЕРАТУРА**

1. Блог системи управління проектами Worksection: [Електронний ресурс]. – Режим доступа: [https://worksection.com](https://worksection.com/)
2. Хабр: [Електронний ресурс]. – Режим доступа: [https://habr.com](https://habr.com/)
3. Методологія управління проектами: [Електронний ресурс]. – Режим доступа: [https://netology.ru](https://netology.ru/)
4. ScrumTrek: [Електронний ресурс]. – Режим доступа: [https://scrumtrek.ru](https://scrumtrek.ru/)
5. HappyMonday Talks: online-журнал про кар'єру і пошук можливостей: [Електронний ресурс]. – Режим доступа: <https://talks.happymonday.com.ua/>
6. Генрік Кніберг. SCRUM И KANBAN: Выжимаем максмум / Генрік Кніберг, Матіас Скарін. – Швеція: C4Media Inc., 2010. – 78 с.
7. Вікіпедія: [Електронний ресурс]. – Режим доступа: https://wikipedia.org