**Міністерство освіти і науки України**

**Національний університет «Запорізька політехніка»**

кафедра програмних засобів

**РЕФЕРАТ**

на тему:

«МЕТОД РОЗРОБКИ KANBAN. РІЗНИЦЯ МІЖ KANBAN ТА SCRUM»

Студента групи КНТ-227

С.Р.Медведєва

Викладач

Г.В. Табунщик

2019

# ЗМІСТ

[ЗМІСТ 2](#_Toc22656104)

[ВВЕДЕННЯ 3](#_Toc22656105)

[1. KANBAN ТА SCRUM 4](#_Toc22656106)

[2. ЯК СКЛАДАЮТЬ СПИСОК ЗАВДАНЬ 6](#_Toc22656107)

[3. ЯК ПРАЦЮЮТЬ НАД ПРОЕКТОМ 7](#_Toc22656108)

[4. ЩО ЗА ДОШКИ ВИКОРИСТОВУЮТЬ SCRUM І KANBAN 9](#_Toc22656109)

[5. ЯК ВПРОВАДЖУЮТЬ 10](#_Toc22656110)

[ВИСНОВКИ 11](#_Toc22656111)

[ПЕРЕЛІК ПОСИЛАНЬ 12](#_Toc22656112)

# ВВЕДЕННЯ

Scrum і Kanban - представники методологій Agile-сімейства. Обидві вважаються гнучкими і ітеративним. Перед тим, як розібратися в різниці між ними, згадаємо коротко в тому, що їх об'єднує.

Більше 17 років тому лідери IT-розробки сформулювали маніфест Agile. Головне, що можемо виділити з маніфесту:

* Люди і взаємодія важливіше процесів та інструментів.
* Працюючий продукт важливіше вичерпної документації.
* Співпраця з замовником важливіше узгодження умов контракту.
* Готовність до змін важливіше проходження попереднім планом.

Можна сміливо погодитися з усіма аргументами, адже дійсно:

* Люди важливіше інструментів, тому що декілька людей, об'єднаних разом і «заряджених» на одну спільну мету можуть мати більший потенціал і прийти в підсумку до більшому результату.
* Слово «замовник» дуже проситься поміняти на «користувач». Вимоги до проекту треба збирати не у замовника, а у користувачів майбутнього продукту.

1. **KANBAN ТА SCRUM**

Основу Scrum складають короткі ітерації або спринти, як правило, 2-3-х тижневі. Перед початком спринту команда сама формує список можливостей на ітерацію, далі запускається спринт.

Після закінчення спринту виконані можливості заливаються на продукцію, а невиконані - переносяться в інший спринт. Як правило, можливості, які робляться під час спринту, не змінюються: що було на старті спринту - має бути зроблено за всяку ціну до закінчення спринту.

На Kanban ми подивимося туди, де він виник. Уявіть собі конвеєр, на якому роблять деталі для машин Toyota. Є верстат, він робить дзеркала для машин. Він вміє робити ліві дзеркала, праві дзеркала, задні і дзеркала для сонцезахисного козирка. Принцип простий: натисни на кнопку, поміняй режим - отримай нову продукцію.

Ось ви замовляєте нову Toyota Camry, і для вас уже роблять дзеркала в козирку. Важливий момент тут - ми можемо міняти пріоритети в будь-який момент. Ми дуже швидко можемо перемикати верстат в інший режим.

Основна різниця між Scrum і Канбан - в довжині ітерацій. У Scrum ітерації - 2 тижні, в Kanban завдання програмісту можна «підсовувати» хоч кожен день.

Kanban дає більше гнучкості, якщо під гнучкістю розуміти частоту зміни пріоритетів. Вчора ви залили на продукцію нову можливість, а сьогодні отримали дані з передової і дізналися, що ось ця штука не працює так, як було задумано - люди не натискають кнопку «купити». Ви «даєте по шапці» UX, він дає вам нові вимоги. Ви піднімаєте нагору черзі цю задачу, програміст бере цю задачу «зверху», виконує її і, до вечора fix вже на продукції, конверсія в платежі зросли на 12%. Це перемога.

У Scrum завдання прийнято оцінювати в Story points або в годинах. Без оцінки не вийде сформувати спринт: адже нам потрібно знати, чи встигнемо ми зробити завдання за 2 тижні. Через 2 тижні ми отримуємо цінну статистику - скільки годин або Story points команда змогла зробити за спринт. Velocity - це продуктивність команди за один спринт. Цей параметр дозволяє Scrum менеджеру передбачити, де команда буде через 2 тижні.

У Kanban не прийнято робити оцінку. Це опціонально, команда вирішує сама. Тут немає поняття «швидкість роботи команди», вважається тільки середній час на завдання. Час це вважається за допомогою спеціального звіту - Cycle Time.

Cycle Time для завдання = час виконання завдання мінус час початку роботи над завданням. Наприклад, у вас є колонки: to do, reopened, developing, testing, stage testing, deployed. Cycle time для завдання буде дорівнює deployed-developing, тобто скільки часу пройшло від моменту, коли завдання почалось робитись до моменту поки воно потрапило в deployed.

Отже, в Scrum наша мета - закінчити спринт, в Kanban - завдання.

Scrum - це автобус, який зупиняється лише на певних зупинках, де люди виходять групами. А Kanban - це маршрутка: захотів пасажир вийти, попросив водія і вийшов там, де йому потрібно.

1. **ЯК СКЛАДАЮТЬ СПИСОК ЗАВДАНЬ**

Для початку команда бере проект і ділить його на десятки, а то і сотні завдань поменше. Це частина філософії Agile, тому так роблять і в Kanban, і в Scrum.

Всі завдання проекту, які належить виконати, складають у загальний список - беклог. Беклог - це банк завдань проекту. Кожне завдання має бути актуальна. Якщо буде потрібно, їх можна додавати в беклог або видаляти з нього «на льоту».

Кожне завдання має вагу - зазвичай це час, який потрібно на рішення. Команда сама оцінює вагу всіх завдань, тому якщо проект не закінчений в строк, винна команда.

Також у кожного завдання є пріоритет. У Kanban пріоритети розставляє команда, в Scrum - власник продукту. Пріоритети можна і навіть потрібно переглядати по ходу проекту - це один із стовпів гнучких методологій.

1. **ЯК ПРАЦЮЮТЬ НАД ПРОЕКТОМ**

Коли починається робота, Scrum стає зрозуміло, чому його називають набагато більш директивної методологією.

Scrum-команда розбиває час роботи над проектом на рівні відрізки - спринти. Спринт може тривати і день, і місяць, а в останні роки стандартом став спринт в 2 тижні.

Оскільки все спринти однакові за тривалістю, в роботі команди з'являється ритм. Ритм - важливий аспект методології.

Спринт складається з чотирьох послідовних етапів:

* **Планування.** Команда перевіряє завдання в беклогах і вибирає найбільш пріоритетні. На спринт беруть стільки завдань, скільки встигнуть зробити.
* **Виконання.** В ідеальній команді фахівці працюють паралельно: поки програміст створює код, тестувальник пише до нього тести, а технічний письменник - документацію.
* **Реліз.** Команда представляє результати своєї роботи світу. До моменту кожного релізу продукт повинен бути працездатним, корисним для користувача і більш досконалим, ніж до спринту.
* **Ретроспектива.** Команда обговорює спринт і можливі проблеми. Всі разом думають, як поліпшити роботу і зробити в наступному спринті більше.

У Scrum категорично не можна додавати завдання в поточний спринт, тому Scrum менш гнучкий. Навіть якщо з'явилася термінова і важлива задача, вона піде в роботу тільки з наступного спринту.

В кінці спринту недороблені завдання йдуть назад в беклог. Чи потрібно їх доробляти і коли, визначають на етапі планування наступного спринту.

У порівнянні зі Scrum-директивами Kanban - це оплот лібералізму і хаос: спринтів як таких немає. Проект зазвичай ділять на ітерації, але вони можуть бути будь-якої довжини. Ритмічність Kanban не наказує.

Згадаймо етапи scrum-спринту:

* **Планування.**
* **Виконання.**
* **Реліз.**
* **Ретроспектива.**

У Kanban одні етапи не пов’язані з іншими. Настають вони, коли вирішить команда. Наприклад, реліз - по вівторках, планування - коли закінчилося завдання, ретроспективи - кожен останній четвер місяця, а на тлі всього цього безперервно йде розробка.

Оскільки виражених спринтів немає, з'являються особливості:

* Нові завдання додають в будь-який час. Якщо потрібно терміново щось зробити, команда не чекає наступного спринту;
* Задача залишається в роботі скільки завгодно довго, поки команда не закінчить її або не скасує.

1. **ЩО ЗА ДОШКИ ВИКОРИСТОВУЮТЬ SCRUM І KANBAN**

Дошка - це серце kanban- і scrum-розробки, єдиний візуальний атрибут методологій. Дошки насамперед вішають на стіну, коли хочуть показати, що працюють по Kanban або Scrum.

Дошку використовують, щоб зробити проект більш прозорим, розпланувати завдання і поставити обмеження.

Дошку розкреслюють на стовпці. Кожен стовпець - це стан завдання ( «Розробка», «Тестування», «Реліз»). Кількість стовпців залежить від проекту, але чим їх менше, тим краще.

Картки - це завдання. На кожній опис, вагу і пріоритет. Коли завдання проходить черговий етап, його переклеюють в відповідний стовпець. При простому погляді на дошку зрозуміло, як справи з проектом в цілому і з кожним завданням.

Дошки бувають фізичні і електронні.

Фізична дошка. Часто дошка - це дійсно дошка з клітинами з малярного скотча. Завдання - липкі листочки, які зручно рухати по дошці.

Справжня дошка знаходится в кімнаті і кожен в команді бачить, як йдуть справи. Навколо цієї дошки добре збиратися під час нарад.

Електронна дошка. Електронна дошка під рукою на пляжі, на дачі, в машині і в метро. Якщо у вас є співробітники, які працюють віддалено, електронна дошка - єдиний вихід.

1. **ЯК ВПРОВАДЖУЮТЬ**

Перейти до Agile простіше через Kanban. У цій методології менше обмежень, можна не формувати крос-функціональні команди, а як і раніше ділитися на відділи. Scrum можна спробувати, коли відчуєш суть гнучкої методології.

1. Знайдіть настільну гру getkanban і пограйте в неї з колегами. Офіційна версія коштує $ 450, але в інтернеті є й інші варіанти.
2. Якщо гра сподобалася і ви відчули методологію, візьміть будь-який проект і поділіть його на дрібні завдання. Зручно, коли всі завдання приблизно рівні за часом виконання. Призначте пріоритет для кожної.
3. Подумайте, які стадії проходить будь-яке завдання. У найпростішому варіанті це може бути «Заплановане → В роботі → Завершено». Варіант складніше: «Заплановане → Дизайн → Розробка → Інтеграція → Готово». Статуси залежать від того, що саме ви робите.
4. Проставте обмеження на статус завдань так, як підказує логіка. Якщо у вас два дизайнера і три програміста, нехай в колонці дизайну буде максимум 2 завдання, а на дошці «В розробці» - 3.
5. Проведіть першу нараду. Обговоріть, як організуєте роботу. Коли будете реалізовувати продукт, а коли збиратися на наради.
6. 6.Перенесіте кілька завдань з беклога в «Заплановано» і запускайте процес.
7. Слідкуйте, щоб картки подорожували між колонками, коли завдання перейшло з одного статусу в інший.
8. Заміряйте середній час виконання завдання. Додаючи картку на дошку, пишіть на ній час початку роботи, а знімаючи - час завершення.
9. Експериментуйте з обмеженнями і робочим процесом так, щоб середній час виконання завдання зменшувалася.
10. Збирайтеся на ретроспективні зібрання, де ви обговорюєте роботу над проектом: що пройшло добре, що виходить погано, як можна поліпшити роботу.

# ВИСНОВКИ

Підбиваючи підсумки, хочу зазначити, що Scrum - гнучка методика розробки, а Kanban - ще більше. Всьому свій час і місце. Якщо це розробка нового продукту, то на старті розробки і до релізу краще використовувати Scrum, так як він робить розробку більш контрольованою за термінами. Також в Scrum багато комунікацій в команді: хлопці обговорюють весь беклог спринту перед стартом, задають питання авторам завдання (UX, продакт-менеджерам, бізнес-аналітикам), оцінюють завдання спільно з допомогою Planning poker. Scrum допомагає детально занурити команду в суть продукту.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Scrum vs Kanban: в чем разница и что выбрать? [Электрон. ресурс]. – Режим доступа: <https://habr.com/ru/company/hygger/blog/351048/>.

2. Разбираемся в Scrum и Kanban: [Электрон. ресурс]. – Режим доступа: <https://netology.ru/blog/scrum-kanban>.