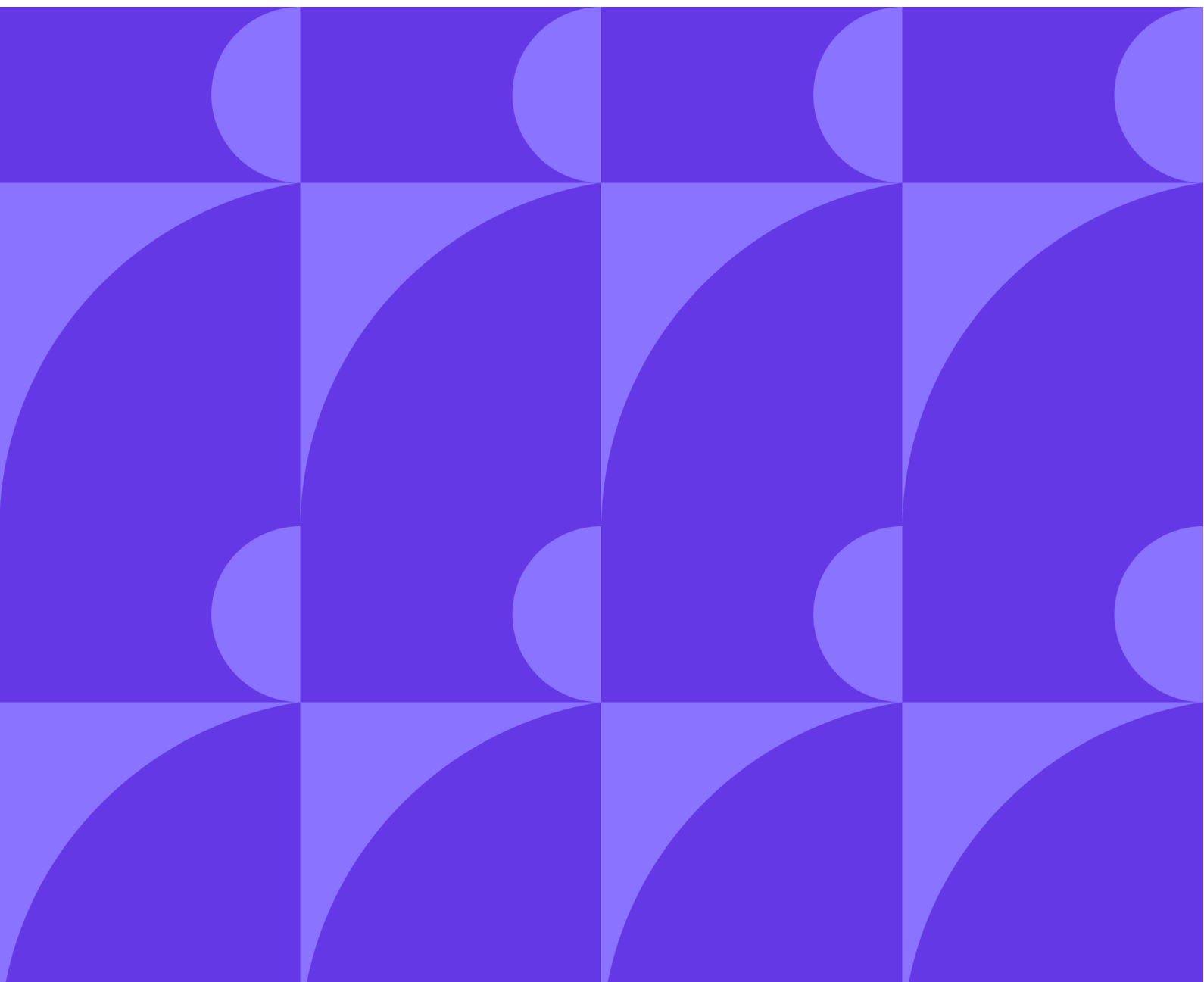


 Гибкие методологии

# Kanban



# На этом уроке

1. Узнаем, что такое Kanban, и рассмотрим историю его возникновения.
2. Поговорим про особенности методологии.
3. Рассмотрим, как по Kanban строится работа команды.
4. Разберём плюсы и минусы Kanban.

## Оглавление

На этом уроке

Оглавление

Глоссарий

Теория урока

История возникновения Kanban

Что такое Kanban

Роли SDM и SRM

Практики Kanban

Принципы Kanban

Kanban, Scrum и Agile. Где связь?

Как внедрить Kanban?

Используемые источники

## Глоссарий

**Каденция** — проверка или совещание, предоставляющее обратную связь по сервису. Временной период между проверками.

**Карточка** — визуализация рабочего элемента.

**Скорость поставки** — количество рабочих элементов, выходящих из системы по завершении работ в единицу времени. Другое название — скорость завершения.

**Kanban-доска** — доска с карточками задач. Наглядно показывает объём работы.

**Kanban-митинг** — ежедневное собрание у канбан-доски. Основная социальная составляющая метода и механизм получения обратной связи. В центре внимания — жизненный цикл разработки.

**Вытягивающая система** — система планирования и поставки работ при наличии нагрузки и свободных мощностей. Работа над элементом начинается только при наличии соответствующего запроса и мощностей в системе. Kanban — это вытягивающая системой.

**Критерии вытягивания** — определение того, когда работа в рамках определённой активности считается выполненной, и рабочий элемент может двигаться дальше.

**Стоимость задержки** — недополученная прибыль или понесённые издержки из-за не вовремя оказанных услуг.

## Теория урока

1. История возникновения Kanban
2. Что такое Kanban
3. Роли SDM и SRM
4. Практики Kanban
5. Принципы Kanban
6. Ценности Kanban
7. Kanban, Scrum и Agile. Где связь?
8. Как внедрить Kanban

## История возникновения Kanban

Kanban как термин в управлении появился в компании Toyota. В переводе с японского означает «сигнал» или «карточка». Так называли записки с запросом на какие-то действия.

На автомобильных заводах с помощью карточек передавали информацию о том, сколько и каких деталей потребуется на следующем этапе.

Toyota применила систему Kanban в своих цехах, чтобы лучше соотнести внушительные складские запасы и реально используемые в производстве материалы:

- Для отслеживания объёмов производства в цехе и взаимодействия с поставщиками использовалась карточка (kanban), которую рабочие передавали между командами.
- Когда в корзине заканчивались нужные на участке производства материалы, на склад передавали kanban с запросом на них.
- На складе уже стояла новая корзина с этими материалами: её отправляли в цех, а складские работники отсылали поставщику свой kanban.
- У поставщика корзина с материалом тоже была готова, и он отправлял её на склад.

**Пример:** компании нужно сделать 10 автомобилей *точно в срок*. Можно заранее определить, сколько деталей потребуется на определённых этапах, а затем начать с конца вытягивать нужное их количество, отвечая на вопросы:

- Сколько литров краски потребуется?
- Сколько колёс?
- Сколько двигателей?
- И так далее.

Ответив на вопросы, можно понять, сколько и каких деталей нужно для производства 10 автомобилей в срок. А после этого сделать на них kanban — карточку с пул-запросом.

Таким образом, нет излишка запчастей, а на складах, логистике и других издержках можно сэкономить.

Kanban-метод в IT придерживается понятия **«точно в срок»**, но, в отличие от заводов Toyota, подразумевает интеллектуальный труд.

Код программиста нельзя пощупать и увидеть, пока он не превратится в конечный продукт (ПО с пользовательским интерфейсом). Kanban-метод позволяет визуализировать поток интеллектуальной работы и сокращает количество незавершённых задач. За счёт этого достигается равномерная и предсказуемая скорость разработки ПО.

Kanban как инструмент в IT-сфере был представлен Дэвидом Дж. Андерсоном в компаниях Microsoft (2005) и Corbis. Широкое распространение и название метод получил в 2007 году. В 2010 году Дэвид Андерсон написал книгу «Kanban. Альтернативный путь в Agile», в которой подробно раскрыл сущность метода в разработке ПО.

## Что такое Kanban

**Kanban** — это часть Agile-философии и метод улучшения рабочих процессов путём визуализации и активной работы над незавершёнными задачами.

Kanban — не методология и не фреймворк. Скорее это метод управления или подход, который можно применять, дополняя уже существующий рабочий процесс.

Kanban помогает управлять работой и улучшать предоставление сервисов до такой степени, чтобы постоянно соответствовать ожиданиям клиентов. Это способ улучшить то, что вы и так уже делаете, а не замена текущих процессов.

#### **Внимание!**

Kanban — способ оптимизировать рабочий процесс. Чем бы вы ни занимались, практики Kanban позволят сделать работу лучше, чтобы с большей вероятностью попадать в ожидания заказчика.

## Роли SDM и SRM

Чтобы выстроить процесс работы по Kanban, нужна команда и менеджеры. Kanban предполагает несколько ролей в команде разработки помимо самих разработчиков:

### Service Request Manager (SRM)

Service Request Manager (SRM) — менеджер запросов для сервиса. Несёт ответственность за:

- изучение потребностей и ожиданий заказчиков,
- выбор и приоритизацию рабочих элементов на *собраниях по пополнению*.

Другие названия роли — менеджер продукта, владелец продукта, менеджер сервиса.

В материалах по Kanban авторы изредка говорят, что эта роль нужна скорее, чтобы не увольнять product owner при переходе со Scrum. В позиции SRM нет нужды, если нет проблем с получением заданий от заказчика.

Роль может быть полезна тем командам, которые страдают от отсутствия «единой точки входа» в продуктовых компаниях. Тогда реквест-менеджер может устанавливать формальные политики и разбираться с входящей очередью. Но это точно не работа на полную ставку. Как вариант, один реквест-менеджер может работать с несколькими командами. В него можно превратить бывшего менеджера проектов или руководителя группы разработки (того, кто в новом процессе остался без лишней работы).

### Service Delivery Manager (SDM)

Service Delivery Manager (SDM) — менеджер поставки для сервиса. Несёт ответственность за:

- поток работы, в ходе которого выбранные рабочие элементы поставляются заказчиком,

- проведение канбан-митингов и *собрания планирования поставки*.

Другие названия роли — менеджер потока, менеджер поставок и мастер потока.

Service Delivery Manage — гораздо более сильная роль. Часто её выполняет менеджер проектов — тот, кто изначально нужен, чтобы дела доводились до конца.

Если есть деливери-менеджер, реквест-менеджер часто не нужен. Хотя формально они вместе обеспечивают двоевластие: реквест-менеджер занимается задачами с неопределённым статусом (не решено, делать их или нет), деливери-менеджер — теми, которые уже точно нужно делать и доводить до конца.

---

## Практики Kanban

Теперь разберёмся, как по Kanban строится работа команды.

Чтобы оптимизировать процесс разработки IT-продукта, Kanban предполагает шесть практик:

1. Визуализируй.
2. Ограничивай количество незавершённой работы.
3. Управляй потоком.
4. Делай правила работы явными.
5. Внедряй циклы обратной связи.
6. Улучшайся совместно, эволюционируй на основе экспериментов.

Рассмотрим, как применять их в работе.

### 1. Визуализируй

Для визуализации есть специальный инструмент — **канбан-доска**. Это рабочее пространство для большой команды со списком задач, распределённых по приоритетности, этапам выполнения, тегам и другим показателям.

Типичная канбан-доска состоит из трёх колонок с карточками:

1. задачи в списке ожидания (нужно сделать в скором времени);
2. задачи в работе (над ними кто-то трудится прямо сейчас);
3. выполненные задачи (бэклог для аналитики).

Доска может выглядеть и иначе — всё зависит от целей и специфики команды.

Цели доски:

- ускорить процесс перехода задач из категории «надо сделать» в категорию «сделано»;
- быстро выявить слабые места команды и понять, на каких задачах она застревает;
- сократить список задач, одновременно находящихся в работе, чтобы повысить продуктивность команды и качество её работы.

Задачи записываются на карточках, чтобы их было удобно переносить из одной части доски в другую. Внешний вид карточки — ещё один важный аспект визуализации.

Сегодня можно обойтись без физической доски, её успешно заменят приложения. Популярные среди них:

- Trello
- Asana
- Jira
- MeisterTask
- KanbanFlow
- Toggl Plan
- YouTrack

### **Пример работы с канбан-доской**

Разделите доску на три столбца:

1. «Не начато»
2. «В работе»
3. «Готово»

## Канбан-доска = 3 колонки с карточками

### Предложение

- задачи, которые находятся в списке ожидания (их нужно сделать в скором времени)

### Разработка

- задачи в процессе работы (над ними кто-то трудится прямо сейчас)

### Приемка

- выполненные задачи (это бэклог для дальнейшей аналитики)



Запишите задачи на стикеры и приклейте их в столбец «Не начато». Когда начнёте работу над задачей, переместите стикер в столбец «В работе». В какой-то степени Канбан основывается на гонке стикеров — кто быстрее дойдёт до последнего столбца «Готово».

## 2. Ограничивай количество незавершённой работы

Работа в процессе (с англ. Work in Progress или WIP) — показатель количества задач, выполняемых в определённое время.

Канбан формирует понимание, что эффективные системы больше сфокусированы на потоке работы, чем на использовании рабочих ресурсов. Из этого следует, что когда ресурсы используются максимально, в системе нет свободного места. Это приводит к ухудшению потока, как в час пик на автостраде.

Кроме того, в интеллектуальной работе есть проблема переключения контекста, которая может резко снизить эффективность сотрудников.

В Канбан работу в процессе ограничивают, чтобы сбалансировать использование ресурсов и при этом обеспечивать поток работы.

### ⚠ Внимание!

Чтобы достичь успеха в Канбан, нужно вести наблюдения, ограничивать и оптимизировать объём незавершённой работы. Это приведёт к сокращению времени производства, повысит качество и увеличит скорость поставки продукта.

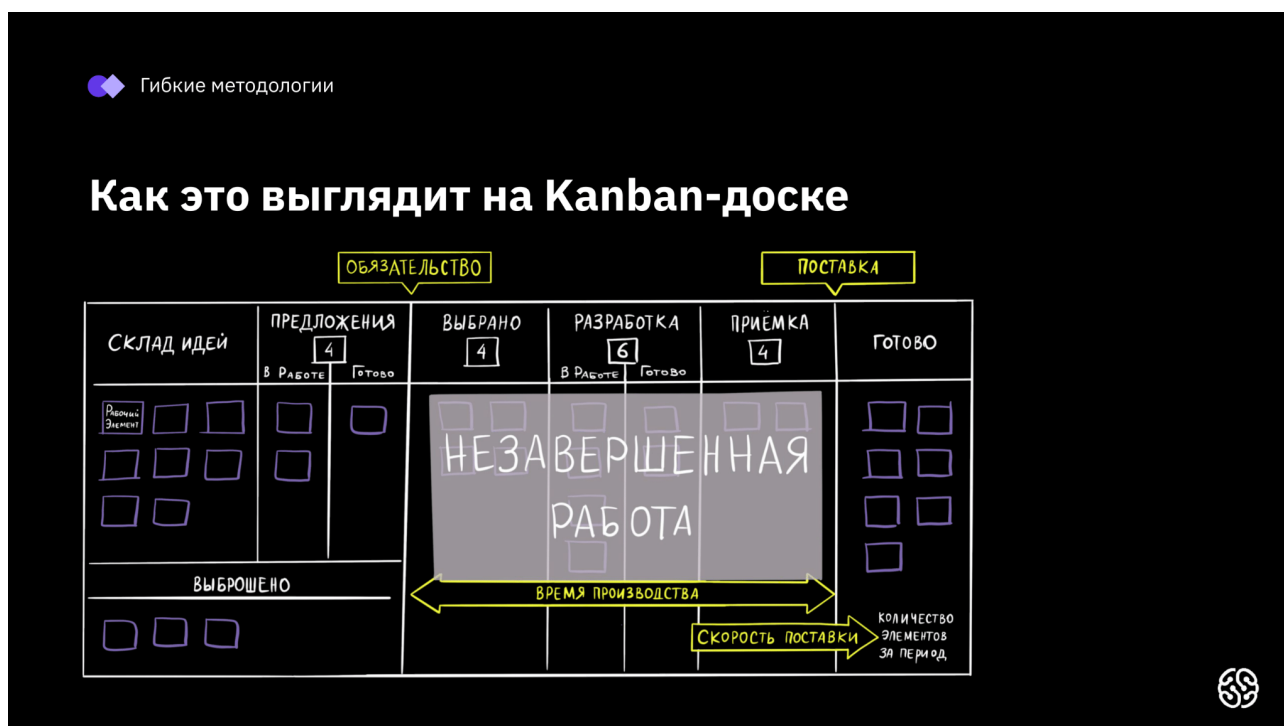


### 3. Управляй потоком

Цель управления потоком в том, чтобы выполнять работу равномерно и предсказуемо, при этом сохраняя устойчивый темп. Это хорошо соотносится с предыдущей практикой — «Ограничивай количество незавершённой работы».

Ограничение работы в процессе — один из ключевых способов, который помогает нам обеспечить плавный и предсказуемый поток такой работы.

Мониторинг или измерение рабочего процесса помогает получить важную информацию, полезную для управления ожиданиями клиентов, прогнозирования и усовершенствования процесса.



На доске изображена поточная система, рабочие элементы которой проходят разные этапы обработки в направлении слева направо. Чтобы такая поточная система стала канбан-системой, должны выполняться условия:

1. В системе должны присутствовать сигналы (как правило, визуальные), ограничивающие количество незавершённой работы (WiP) — это максимальное количество работ, которое может находиться в столбце. В примере источник таких сигналов — карточки с WiP-лимитами (в прямоугольниках в заголовках столбцов) и столбцы активностей.
2. В канбан-системе должны быть определены точка принятия обязательств и точка поставки.

Обязательство — это чётко сформулированное негласное соглашение между заказчиком и сервисом о том, что:

- заказчик желает получить рабочий элемент и согласен принять его поставку;
- сервис готов обеспечить производство и выполнить поставку элемента заказчику.

Перед точкой принятия обязательств формируется ряд запросов в очереди (или склад идей), которые могут выбираться в работу или выбрасываться за ненадобностью. Кроме того, формируется процедура выбора элементов из представленных вариантов. Они берутся в работу и выполняются, после чего элемент поставляется заказчику.

## 4. Делай правила работы явными

Это нужно, чтобы сделать процессы доступными для новичков. С явными правилами можно сократить время адаптации нового члена команды разработки.

Kanban предлагает внедрить основные правила в работе:

- Пополнение канбан-доски (когда, сколько, кем).
- Определение критерия готовности, после которого рабочий элемент может двигаться дальше (это называется «критерии вытягивания»).
- Ограничение работы в процессе.
- Правила работы с рабочими элементами в зависимости от их класса обслуживания.
- Время и содержание митингов.
- Другие принципы и договорённости о совместной работе.

Эти правила должны быть согласованы всеми вовлечёнными сторонами, включая клиентов и сотрудников, ответственных за работу с канбан-доской.

Правила работы размещают в хорошо видимом месте, желательно рядом с доской. При необходимости их нужно адаптировать и совершенствовать.

### **Внимание!**

Правила работы — это не инструкции, освобождающие от принятия осмысленных решений. Они должны способствовать самоорганизации участников команды.

Любые создаваемые правила работы должны быть:

- рассредоточенными;
- простыми;
- чётко определёнными;

- видимыми;
- всегда применяемыми;
- легко заменяемыми.

**Пример:** вы разделили доску на три части, расписали задачи и постепенно начали их перемещать из «Не начато» в столбец «В работе». В определённый момент обнаруживаете, что «В работе» скопилось много задач, специалисты не справляются, появляются недочёты и ошибки.

Старайтесь не допускать, чтобы во втором столбце сразу появилось большое количество задач — это мешает продуктивной и сосредоточенной работе над конкретным делом.

Сначала определите WiP-лимиты (максимальное количество задач в работе): например, три, исходя из возможностей команды. Чтобы это было видно всем членам команды, запишите это число «3» над столбцом «В работе».

Затем, когда кто-то из команды захочет взять себе новые задачи из первого столбца «Не начато», он сразу увидит, что столбец «В работе» загружен тремя задачами и переполнять его нельзя. Есть задачи в работе, по которым, возможно, требуется помощь его коллеге.

В примере сразу задействованы несколько практик Kanban:

- ограничивай количество незавершённой работы;
- управляй потоком;
- делай правила работы явными.

## 5. Внедряй циклы обратной связи

Циклы обратной связи нужны для скоординированной поставки и улучшения предоставления вашего сервиса.

Работающий набор циклов обратной связи усиливает способность организации к обучению и эволюции посредством контролируемых экспериментов.

Основные инструменты для циклов обратной связи в Kanban:

- доска,
- метрики (критерии для измерения качества выполненной работы),
- набор регулярных митингов и ревью (*каденции*).

Каденция — термин из музыки. В Kanban так называют регулярные встречи, которые являются петлями обратной связи. Регулярность задаёт ритм, с которым течёт поток работы.

Kanban предполагает 7 каденций.

**Пример:** вы с командой занимаетесь разработкой мобильной игры.

1. **Канбан-митинг** (ежедневно на 15 минут):

Каждый день на 15 минут вы с командой собираетесь у канбан-доски и обсуждаете, что сделали, как проходит ваш поток работы, что можно улучшить в ближайшем будущем (в рамках дня). Уделяете внимание застопорившимся работам, возможным заторам и членам команды, которые ничего не делают для поддержания эффективности рабочего потока. Все члены команды должны проявлять активное участие: чем больше вовлечённых людей, тем больше точек зрения и предложений по улучшению.

К примеру, в вашей команде один из разработчиков уходит от выполнения определённой работы, это заметно изо дня в день. На собрании команда должна выяснить, почему это происходит. Важно, чтобы вопрос задавал не менеджер, а вся команда. Возможно, разработчик перегружен и не успевает, или ему не хватает технических навыков, нужна помощь.

2. **Собрание по пополнению очереди** (каждую неделю на 30 минут):

Нужно определить количество задач, которые будут в работе одновременно, и выделить задачи в работу из бэклога. Так поддерживается постоянный поток задач, движущихся по канбан-доске. Вы с командой выбираете 3-4 задачи из бэклога и перемещаете их на доске из столбца «Не начато» в столбец «В работе», обсуждаете различные моменты и расходитесь, приступая к их выполнению.

3. **Собрание планирования поставки** (зависит от количества поставок промежуточного результата работ, занимает 1-2 часа):

Более глобальная встреча с обсуждением всех моментов выдачи заказчику какого-то результата по проекту. На ней можно определить дату релиза одного из этапов проекта, например, когда вы сможете передать заказчику MVP мобильной игры. Уточните, кто со стороны заказчика и по каким параметрам будет проверять MVP, кто будет участвовать в тестировании, какие типы тестирования будут проводиться.

4. **Ревью сервиса поставки** (раз в две недели на 30 минут):

В этой встрече должны участвовать заказчик (или его представители), менеджер сервиса поставки и представители от команды поставки. Здесь нужно понять и обсудить с заказчиком, насколько он доволен скоростью выполнения работ (пропускной способностью команды). Опросите заказчика, задайте наводящие вопросы, спросите его про желаемый срок выполнения намеченных работ и всего проекта в целом. У вас будут конкретные метрики, по которым можно измерить уровень удовлетворённости заказчика.

5. **Операционная встреча** (раз в месяц на 2 часа):

Здесь выявляются слабые и сильные места в командной работе над всеми проектами в компании. Даже если отдельные команды высокоэффективны,

организация в целом может отставать из-за неправильно функционирующей части или неэффективной передачи работы.

На этой встрече присутствуют менеджеры из разных подразделений, отделов и систем. Они ищут способы повысить эффективность в целом. Особое внимание уделяют взаимозависимости между канбан-системами, которые могут вызывать цепную реакцию по срокам поставки. Например, параллельно с вашим проектом по разработке мобильной игры, компания занимается разработкой сайта для интернет-магазина, а значит есть как минимум две канбан-системы. Над двумя проектами работает один дизайнер. Возможно, он перегружен и отстаёт по срокам, что затягивает процесс выдачи MVP заказчику. Таких сложных переплетений в работе команд на живых проектах много. На операционных встречах выявляются и принимаются решения о том, как улучшить работу взаимосвязанных сервисов.

6. **Ревью рисков** (раз в месяц на 1-2 часа):

Здесь обсуждают риски, которые могут сработать при поставке MVP мобильной игры, смотрят, какие задачи тормозят по срокам, какие сложности возникли в разработке, и как их решить, вырабатывают меры по снижению рисков на разработку, чтобы не просрочить дату поставки MVP заказчику.

7. **Ревью стратегии** (ежеквартально на полдня):

В обзоре стратегии участвует высшее руководство, владельцы портфолио/продукта, члены senior-команды из отделов по работе с клиентами (продажи и маркетинг).

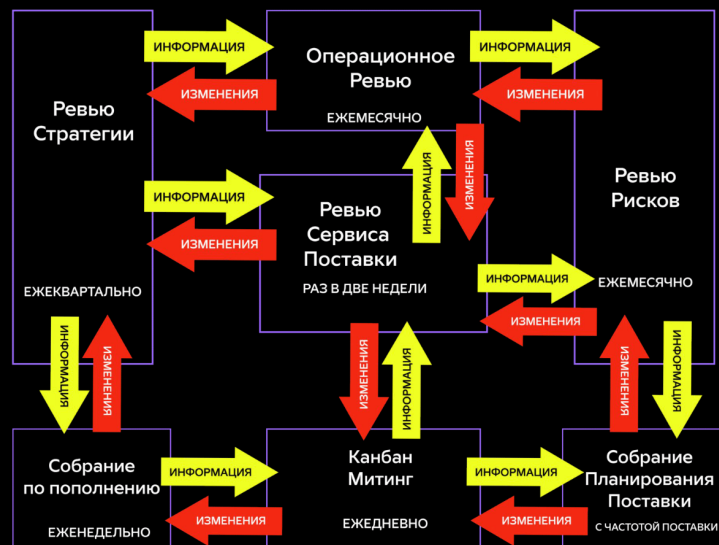
Все вместе они рассматривают операционную деятельность компании в целом. На встрече обсуждают новые тренды в мире разработки, анализируют опыт конкурентов, смотрят, как идёт разработка мобильных игр у других компаний, сравнивают свою скорость поставки со скоростью рыночных изменений.

По итогу таких встреч внедряют новые подходы и практики в работу каждой из команд, где это необходимо, то есть вырабатывают стратегию компании.

Смысл ревью стратегии в том, чтобы выявить потенциальные масштабные проблемы и скорректировать курс или оптимизировать взаимодействие команд в случае необходимости.

Вот так это выглядит на схеме:

## Каденции



## 6. Улучшайтесь совместно, эволюционируйте на основе экспериментов

Эта практика помогает реализовать принцип управления изменениями: мы начинаем с того, что делаем сейчас, и соглашаемся достигать улучшений через эволюционные изменения. Это самое важное в Kanban — *мы ничего не рушим, мы стараемся улучшить имеющуюся систему.*

Kanban — метод непрерывных изменений, и мы вносим эти изменения совместно, используя контролируемые эксперименты, которые основаны на моделях и научном подходе. Чтобы оценить внедрённые эксперименты, мы должны сформировать метрики и получать по ним обратную связь.

### ⚠ Внимание!

Мы разрабатываем безопасные эксперименты так, чтобы при хороших результатах мы могли сохранить изменения, а при недостаточных — легко вернуться к предыдущему состоянию.

Пример: в вашей компании есть проблемы со сроками разработки IT-продукта. Заказчики по разным проектам недовольны и выставляют вам претензии. Компании

приходится платить штрафы, и вам нужно исправить ситуацию. При этом в компании используется Waterfall — каскадная методология управления IT-проектами.

Если бы вы выбрали Scrum, необходимо было бы поменять подход к разработке полностью, сломать всё, что есть, и выстроить процессы заново. Вы бы стали делить большие проекты на малые части и работать спринтами, в команде бы появились новые роли — scrum-мастер и product owner. Возможно, это привело бы вас к успеху и исправило проблему с задержкой сроков разработки, но не факт. После слома старой системы нужно время, чтобы построить новую, отладить процессы и коммуникацию.

Если вы выберете Kanban, не придётся ломать структуру внутри компании и команды. Вы просто начнёте искать причину возникшей проблемы со сроками разработки.

В первую очередь вам нужно будет определить весь бэклог текущих задач на проекте, затем понять, какие задачи в работе, и сколько людей над ними трудится. Возможно, именно здесь вы и найдёте причину срыва сроков — слишком перегружены специалисты и много задач в одновременной работе, их количество нужно ограничить. Наладив всё, через определённое время, вы снимете метрики, проанализируете обратную связь от команды и заказчика и поймёте, что ещё можно доработать и улучшить.

Таким образом, вы реализуете шесть принципов Kanban на практике.

## Принципы Kanban

Шесть принципов Kanban делятся на две группы:

### I. Принципы управления изменениями

1. Начните с того, что есть сейчас: поймите текущий процесс и практики, соблюдайте текущие роли, обязанности и должностные инструкции.
  - так мы сводим сопротивление переменам к минимуму;
  - нужно проявлять уважение к текущей деятельности специалистов, чтобы в дальнейшем вовлекать их в процесс решения сложных задач;
  - текущие процедуры могут нести в себе опыт и гибкость, которые, возможно, специалисты не оценивают из-за того, что выполняют их уже достаточно долго;
  - не ломая текущие процедуры, можно установить точку отсчёта, на основе которой можно в дальнейшем провести анализ эффективности всех внедрённых изменений.
2. Договоритесь между собой относительно того, что будете добиваться улучшений путём эволюционных изменений.

- не нужно навязывать изменения, нужно договориться и все изменения обсудить;
  - нужно менять эволюционно, а не революционно.
3. Поощряйте проявление лидерства на всех уровнях: от рядовых сотрудников до топ-менеджмента.

## II. Принципы предоставления сервисов

1. Выясните потребности и ожидания заказчика и сосредоточьтесь на них.
2. Управляйте работой, а не людьми. Дайте людям возможность организовать себя вокруг неё.
3. Развивайте правила для улучшения бизнес-показателей и увеличения пользовательской удовлетворённости.

Принципы предоставления сервисов предполагают, что нужно сместить акцент на заказчика, конечного пользователя IT-продукта и на ценность IT-продукта. В Kanban важна *клиентоориентированность*. Это — одна из девяти ценностей метода.

Давайте посмотрим, какие ещё ценности несёт в себе Kanban.

## Ценности Kanban

- Прозрачность
- Баланс
- Сотрудничество
- Клиентоориентированность
- Поток
- Лидерство
- Понимание
- Согласие
- Уважение

Рассмотрим эти ценности подробнее.

## Прозрачность



Открытый обмен информацией улучшает поток создания бизнес-ценности IT-продукта. В рамках этого типа ценности используются простые и понятные термины.

## Баланс

Для эффективной работы нужен баланс между разными аспектами, точками зрения и возможностями. Отсутствие баланса между некоторыми аспектами (например, нагрузкой команды и её возможностями) в течение продолжительного времени может нарушить нормальную работу.

## Сотрудничество

Одна из задач Kanban — совершенствование совместной работы. Поэтому сотрудничество представляет одну из основных ценностей.

## Клиентоориентированность

Конечная точка каждой канбан-системы — создание ценности. То есть заказчик должен получить нужный ему продукт или сервис. При этом заказчик остаётся внешним фактором по отношению к сервису, но может являться как внутренним, так и внешним фактором по отношению к организации. Заказчики и ценность, которую они получают, всегда находятся в центре внимания по методу Kanban.

Именно поэтому в Kanban есть понятие *класса обслуживания* (сервиса). По сути, это приоритизация задач. Классы обслуживания нужны, чтобы повысить приоритет определённым типам работ или заказчикам и нивелировать стоимость задержки.

Стоимость задержки — это недополученная прибыль или понесённые издержки из-за не вовремя оказанных услуг.

Рассмотрим влияние стоимости задержки и соответствующий класс обслуживания на примерах:

- Ускоренный класс — неотложная скорая помощь-реанимация. Работа сравнима с каретой скорой помощи: едет по выделенной полосе, у неё нет времени откладывать решение проблемы. Такую работу нужно выполнить как можно скорее.
- Класс с фиксированной датой — стоимость задержки резко возрастает после определённого периода. То есть к определённой дате мы точно должны завершить работу, иначе понесём убытки: например, штрафы.
- Стандартный класс — стоимость задержки растёт пропорционально времени. Если делаем сразу, получаем прибыль сразу. Если делаем долго, получаем прибыль долго.

- Нематериальный класс — делаем, но явной прибыли работа не несёт, а стоимость задержки растёт медленно. Например, уборка в доме. Можно и не убираться регулярно, но через полгода придётся делать генеральную уборку.

Пример: ваша компания для работы использует программу, которая в какой-то момент начала требовать обновление. Задача не срочная, её можно сделать в любой момент, выполнение никак не повлияет на текущий рабочий процесс или результат.

Теперь представим, что задача долго откладывается. Возможно, чтобы обновить программу, нужно было приостановить рабочий процесс, возможно, было просто не до того.

Однажды мы столкнулись с тем, что всё перестало работать, потому что устаревшая программа больше не совместима с другими, которые мы используем. Задача ужалила нас в самое уязвимое место — *стоимость задержки ракетой взлетела вверх: \*\*\*\*\** рабочий процесс остановился, компания не зарабатывает деньги, пока задача не будет выполнена в срочном порядке. Ей сразу же присваивается класс обслуживания «ускоренный». Стоимость задержки мы сможем оценить по потерям, которые понесёт компания, потому что эта задача не была выполнена ранее.

Таким образом, с помощью классов обслуживания мы приоритизируем задачи на канбан-доске. Сделать это можно с помощью цветных стикеров: для ускоренного класса — красный, для фиксированной даты — оранжевый, для стандартного — жёлтый, для нематериального — голубой.

## Поток

Работа представляет собой поток создания ценности (непрерывный или эпизодический). Способность видеть этот поток — отправная точка в использовании Kanban.

## Лидерство

Способность вдохновлять окружающих на действия своим примером, словами и идеями. В большинстве организаций предусмотрена иерархическая структура, при применении Kanban нужно обеспечить лидерство на всех уровнях.

## Понимание

Kanban — метод совершенствования. Понимание того, какая точка является отправной, будет основополагающим. Каждый член команды должен понимать самого себя в целях дальнейшего развития.

## Согласие

Обязательство совместного движения в сторону достижения целей с учётом или урегулированием расхождений во мнениях и различий в подходах. Эта ценность предполагает принятие совместного стремления к совершенствованию.

## Уважение

Высокая оценка и понимание окружающих. Располагается в конце списка, поскольку представляет собой основу, на которой базируются остальные ценности. По сути, все ценности Kanban сводятся к уважению друг к другу.

## Kanban, Scrum и Agile. Где связь?

Kanban — второй по популярности гибкий подход к разработке ПО. Ряд компаний работает одновременно по Scrum и Kanban — получается Scrumban. Скорее всего, в будущем он станет трендом.

При этом Kanban кардинально отличается от Scrum.

Scrum подразумевает жёстко предписанные процессы, которые должны сломать то, что было в компании раньше: «Мы убираем старые шаблоны и начинаем работать спринтами, с утра обязательно проводим стендапы, в конце спринта показываем результат и проводим ретроспективу».

Kanban подразумевает более эволюционные изменения. Его ключевая мысль: *«Меньше незавершённых задач — выше эффективность команды»*.

### **Внимание!**

Agile — это семейство «гибких» подходов к разработке программного обеспечения и целая философия. В её рамках функционируют практически применимые фреймворк Scrum и Kanban-метод.

Scrum — это «подход структуры». Над каждым проектом работает универсальная команда специалистов, к которой присоединяется еще два человека: владелец продукта и scrum-мастер. Первый соединяет команду с заказчиком и следит за развитием проекта, это не формальный руководитель команды, а скорее куратор. Второй помогает первому организовать бизнес-процесс: проводит общие собрания, решает бытовые проблемы, мотивирует команду и следит за соблюдением фреймворка Scrum.


Scrum-подход делит рабочий процесс на равные спринты. Обычно это периоды от недели до месяца, в зависимости от проекта и команды. Перед спринтом формулируются задачи, в конце обсуждаются результаты, после чего команда начинает новый спринт. Спринты

удобно сравнивать между собой, что позволяет управлять эффективностью работы.

Kanban — это «подход баланса». Его задача — сбалансировать разных специалистов внутри команды и избежать ситуации, когда дизайнеры работают сутками, а разработчики жалуются на отсутствие новых задач.


Вся команда едина. В Kanban нет ролей владельца продукта и scrum-мастера. Бизнес-процесс делится не на универсальные спринты, а на стадии выполнения конкретных задач: «Планируется», «Разрабатывается», «Тестируется», «Завершено» и другие.

Главный показатель эффективности в Kanban — среднее время прохождения задачи по доске. Задача прошла быстро — команда работала продуктивно и слаженно. Задача затянулась — надо думать, на каком этапе и почему возникли задержки, и чью работу надо оптимизировать.

 Гибкие методологии

## Scrum и Kanban

Scrum	Kanban
<ul style="list-style-type: none"><li>• Подход структуры</li><li>• Над каждым проектом работает универсальная команда специалистов: владелец продукта, Scrum-мастер и разработчики</li><li>• Делит рабочий процесс на равные спринты</li></ul>	<ul style="list-style-type: none"><li>• Подход баланса</li><li>• Вся команда едина — в Kanban нет ролей владельца продукта и Scrum-мастера</li><li>• Бизнес-процесс делится на стадии выполнения конкретных задач: «Планируется», «Разрабатывается», «Тестируется», «Завершено»</li><li>• <b>Главный показатель эффективности — это среднее время прохождения задачи по доске</b></li></ul>



Scrum и Kanban — практические инструменты Agile-философии. Но у них есть различия:

- Scrum требует фиксированных ролей, у Kanban нет строго необходимых ролей.
- Scrum основан на итерациях, объединяющих планирование, оптимизацию процессов и выпуск. В Kanban это можно делать регулярно или каждый раз, когда вам нужно.
- Команда Scrum требует оценки своей работы, а команде Kanban это не нужно.

 **Внимание!**

Scrum — фреймворк с жёсткими правилами, от которых нельзя отходить. Kanban — лишь инструмент, позволяющий оптимизировать работу. Его можно использовать как отдельно, так и в рамках фреймворков типа Scrum. Даже опираться на уже существующие варианты использования не обязательно — разобрался в принципах работы инструмента и пользуешься.

## Как внедрить Kanban?



Для этого есть практическое пошаговое руководство — STATIK (с англ. The Systems Thinking Approach To Introducing Kanban — Системный подход для представления Канбана).

Включает в себя следующие шаги:

### 1. Определите источники неудовлетворённости.

- Чем недовольны люди, которые участвуют в процессе предоставления сервиса?
- Чем недовольны клиенты?
- Все эти источники неудовлетворенности создают мотивацию к изменениям, что, в свою очередь, является ключом к успешной канбан-инициативе.

### 2. Проанализируйте спрос.

- Что запрашивают клиенты и по каким каналам?
- Какие виды работ и шаблоны наиболее востребованы?
- Получение этой информации — залог создания полной картины, отображающей, какая работа попадает в систему. Помните: управляйте работой, а не работниками!

### **3. Проанализируйте возможности системы.**

- Каковы возможности системы в отношении того, насколько удовлетворяются потребности клиента, какого типа, как быстро и предсказуемо?
- Для этого шага обычно требуются исторические данные для детального анализа.

### **4. Смоделируйте рабочий процесс.**

- Через какие действия проходит каждый из выявленных типов рабочих элементов? Эти действия могут быть последовательными, параллельными или в произвольном порядке. Позже они станут основой для определения столбцов на канбан-доске.

### **5. Определите классы обслуживания.**

- Ответьте на вопрос: как элементы попадают в систему и обрабатываются в ней?

### **6. Разработайте канбан-систему.**

- Канбан-система разрабатывается на основе всех выводов, полученных на предыдущих этапах. Обычно состоит из доски и тикетов, а также других важных элементов: метрик, каденции и правил работы.

## **Используемые источники**

1. Дэвид Андерсон «Kanban. Альтернативный путь в Agile»
2. [Что такое Kanban?](#)
3. [Что такое Канбан-метод – максимально коротко](#)
4. [6 примеров реального применения Канбан в российских компаниях](#)