

Командная разработка с использованием Agile подхода и фреймворка Scrum.

Scrum, Kanban и Scrumban



На этом уроке

1. Рассмотрим структуру семейства Agile.
2. Ознакомимся с историей возникновения и смыслом метода Kanban.
3. Рассмотрим самый популярный фреймворк — Scrum.
4. Сравним подходы и поймём, почему так популярен Scrumban.

Оглавление

[Структура семейства Agile](#)

[Kanban](#)

[История возникновения](#)

[Kanban как гибкий метод разработки ПО](#)

[Принципы Kanban](#)

[Ценности Kanban](#)

[Роли в Kanban](#)

[Преимущества Kanban](#)

[Scrum](#)

[История создания Scrum](#)

[Определение Scrum](#)

[Основные характеристики Scrum](#)

[Элементы фреймворка и его организация](#)

[Основные принципы Scrum](#)

[Ценности Scrum](#)

[Преимущества Scrum](#)

[Применение Scrum](#)

[Сравнение Scrum и Kanban](#)

[Scrumban](#)

[Заключение](#)

[Глоссарий](#)

[Дополнительные материалы](#)

[Используемые источники](#)

[Практическое задание](#)

На первом уроке мы разобрались, что такое проект, какими бывают проекты и как от этого зависит выбор концепции, по которой будет организована работа команды. Теперь мы знаем, что представляет собой Agile, какую философию и какие принципы он транслирует.

Второй урок начнём с рассмотрения структуры Agile-семейства и познакомимся с самыми популярными методами, которые в него входят: Scrum, Kanban и Scrumban.

Структура семейства Agile

Agile, или гибкие методологии разработки (англ. Agile Software Development), — это обобщающий термин для целого ряда подходов и практик, основанных на ценностях Agile-манифеста. Эти подходы называют фреймворками, методами, методиками или гибкими методологиями и на инфографиках изображают в виде семейства. Можно схематично изобразить Agile в виде зонтика, под которым «прячутся» гибкие методологии и фреймворки:

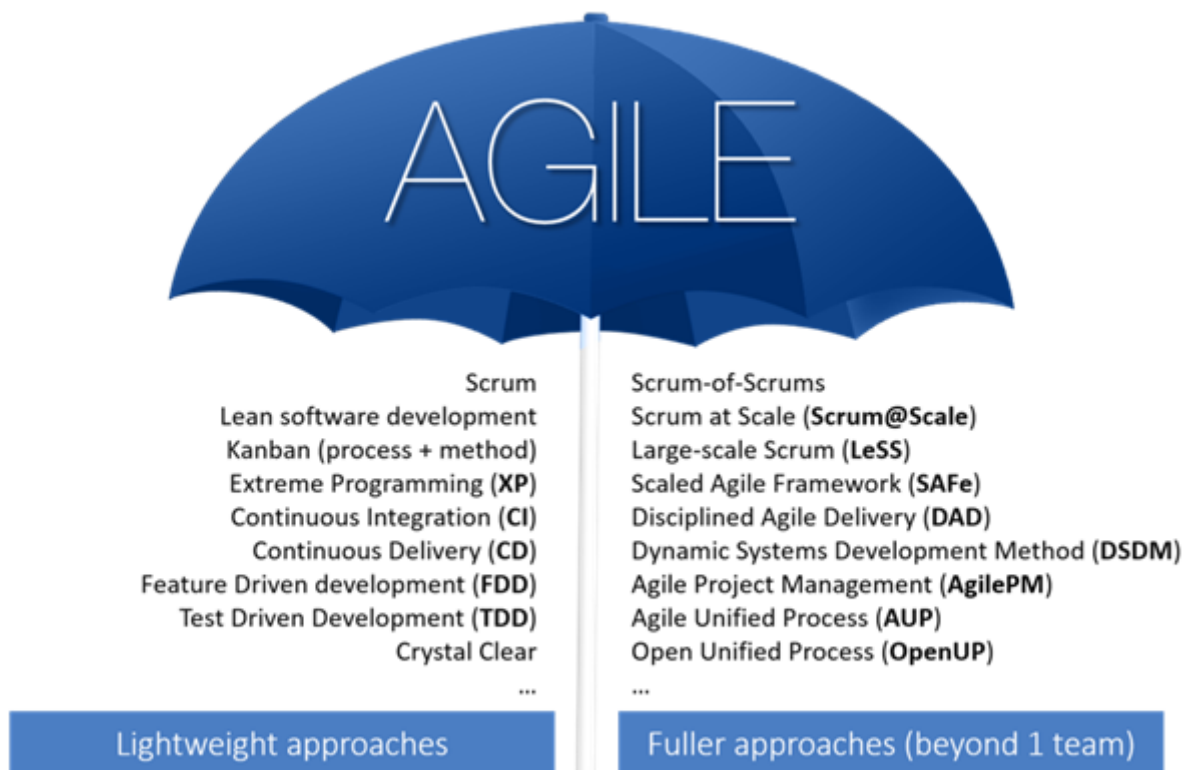


Рис. 1. Семейство Agile

Проект State of Agile — один из самых авторитетных источников информации по применению Agile в мире. Согласно обзору, проведённому на сайте проекта в 2019 году, лидирующие позиции по частоте использования занимал фреймворк Scrum — 54%, на втором месте оказалось использование различных смешанных методов — 14%, на третьем месте был гибрид ScrumXP — 10%, на четвёртом Scrumban — 8%, а на пятом чистый Kanban — 5%.

AGILE METHODS AND PRACTICES

Agile Methodologies Used

Scrum and Scrum/XP Hybrid (64%) continue to be the most common agile methodologies used by respondents' organizations.

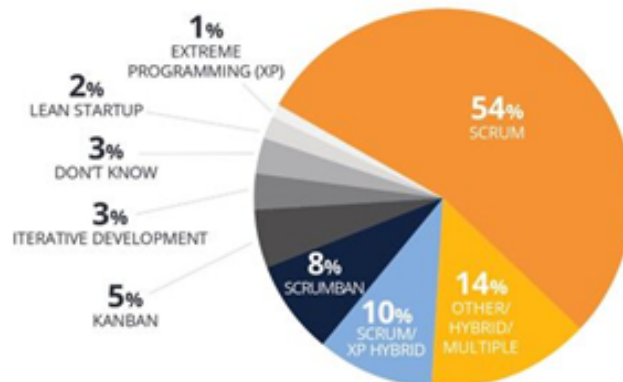


Рис. 2. Соотношение используемых в мире Agile-подходов (2019 год)

В 2020 году ситуация несколько изменилась. Согласно последнему исследованию, самым популярным по-прежнему остаётся Scrum — 58%, на второе место вышел Scrumban — 10%, третье место занимают различные смешанные методы — 9%, на четвёртое место ушёл гибрид ScrumXP — 8%. Kanban по-прежнему на пятом месте, но стал чуть популярнее, прибавив 2%.

AGILE METHODOLOGIES USED

Scrum and related variants continue to be the most common Agile methodologies used by respondents' organizations.

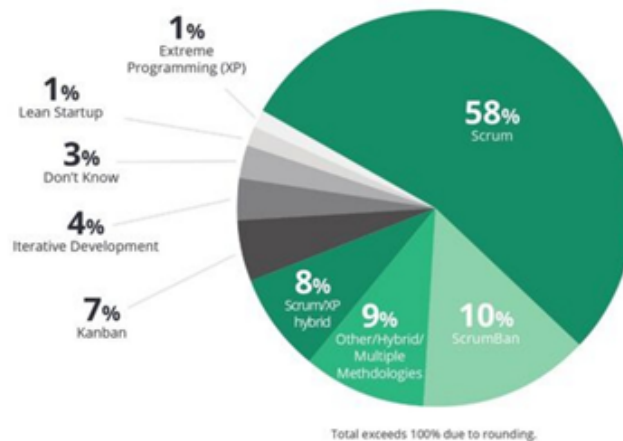


Рис. 3. Соотношение используемых в мире Agile-подходов (данные 2020 года)

На этом уроке мы рассмотрим метод Kanban, постараемся понять, чем так хорош фреймворк Scrum и почему вторым по популярности стал гибрид этих двух методов Scrumban.

Kanban

История возникновения

Термин Kanban (яп. カンバン — «камбан», в управленческой среде устоялся вариант с ошибочной транскрипцией японского слова — kanban) означает «визуальный сигнал, карточка, вывеска». Метод Kanban был разработан и впервые в мире реализован компанией Toyota в рамках её производственной системы. В 1959 году Toyota начала эксперименты с системой Kanban и в 1962 году запустила процесс перевода всего производства на этот принцип.

Работа по системе Kanban помогла компании Toyota реализовать один из главных её производственных принципов — создание потока единичных изделий «Точно в срок». Он предполагает синхронную поставку необходимого в производстве материала: поступление непосредственно в производство на рабочее место к необходимому времени, в необходимом количестве, с предписанным качеством и в соответствующей потреблению упаковке. Следование этой концепции позволило Toyota уйти от операций по обслуживанию производственных запасов и избежать перепроизводства, что, в свою очередь, снижало количество дефектных изделий, простоей людей и оборудования.

Концепция «Точно в срок» включает в себя идею американского первопроходца в области качества Эдварда Деминга: в любой бизнес-системе первоочерёдная задача всех сотрудников организации — удовлетворение и предугадывание желаний потребителя. Эдвард Деминг расширил понятие «потребитель», включив в него не только внешнего потребителя, но и внутреннего. Любой человек и любая стадия производственного процесса должны рассматриваться как потребитель, которого нужно обеспечить всем необходимым в нужный момент. Так появился принцип Деминга «Следующий процесс — ваш потребитель» (по-японски atokotei wa o-kyakusama). Это один из самых важных принципов концепции «Точно в срок». Этот принцип означает: предшествующий процесс всегда должен делать то, чего требует следующий процесс.

К этому принципу присоединилась система вытягивания, в основу которой положен принцип работы американских супермаркетов. В любом хорошем супермаркете запасы товаров на полках пополняются по мере того, как их разбирают покупатели, то есть по мере их потребления. Применительно к цеху это означает, что изготовление или пополнение запаса деталей на стадии 1 должно осуществляться, когда на следующей стадии 2 израсходован почти весь запас деталей, изготовленных на стадии 1, то есть остался лишь небольшой резервный запас. В системе управления Toyota следующая партия деталей со стадии 1 запрашивается только тогда, когда количество деталей, используемых на стадии 2, сократилось до заданного минимума. Благодаря этой системе детали «вытягиваются» по цепочке: от завершающих стадий производственного цикла к начальным.

Сигналом, что уровень запасов сократился до минимума, и стал Kanban — заказ-наряд в виде карточки, которую стали передавать на предшествующую стадию, чтобы известить о потребности и инициировать производство.

Процесс начинается на сборочном заводе. Когда рабочие Toyota на сборочной линии выбирали детали из контейнеров (амортизаторы, петли, дверные ручки и прочее), они извлекали оттуда карточку Kanban и клали её в «почтовый ящик». Работник, отвечающий за транспортировку материалов, совершая

запланированный обход, забирал карточку и возвращал её вместе с пустыми контейнерами туда, откуда поступают детали, — на завод поставщика. Поставщик держит небольшой запас готовых деталей в резерве, который используется, чтобы наполнить вернувшиеся пустые контейнеры и отправить их обратно на линию сборки во избежание простоя. На резервном складе в этот момент возникала потребность восполнить запасы, поэтому соответствующий Kanban и пустые контейнеры отправлялись в производственный цех, где изготавливаются новые детали. Так от потребителя (сборочного завода) заказы добирались до производственного цеха в виде Kanban.

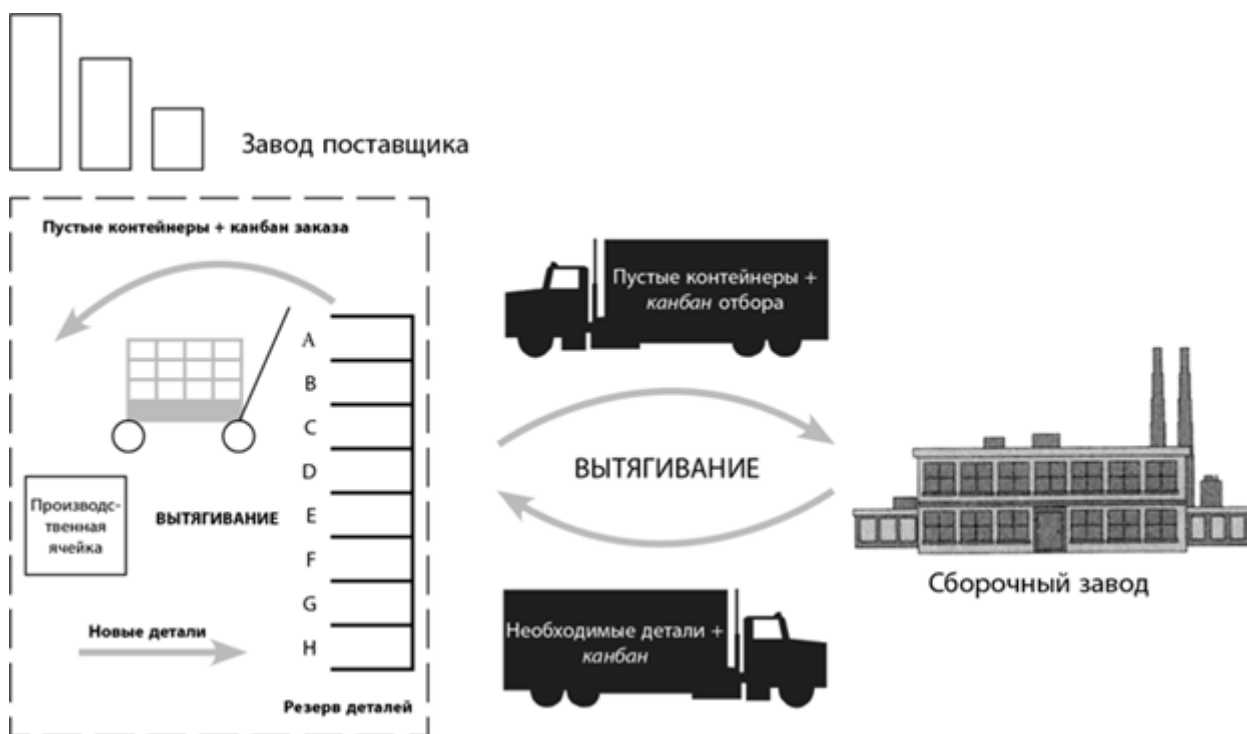


Рис. 4. Схема взаимодействия по принципу вытягивания и циркуляция Kanban-карточек на заводе Toyota

На производственном заводе дополнительно к Kanban-биркам, которые сопровождали контейнеры, появились и Kanban-доски, которые позволяли планировать работу сотрудников на местах.



An original kanban board. Source: Toyota

Рис. 5. Оригинальная Kanban-доска на заводе Toyota

Мастера участков перечисляли выполняемые работы на бумаге и вывешивали их на видном месте рядом с такими же списками мастеров других участков.

Благодаря системе вытягивания и методу Kanban компания Toyota смогла наладить поток единичных изделий «Точно в срок», уйти от операций по обслуживанию производственных запасов и в целом оптимизировать цепочку планирования производственных мощностей, начиная от прогноза спроса, планирования производственных заданий, балансировки и распределения этих заданий по производственным мощностям с оптимизацией их загрузки.

Со временем эта концепция стала универсальной и начала внедряться на производствах по всему миру, доказывая свою эффективность. Метод оказался столь удачным, что привлёк внимание IT-специалистов. Набор практик, который обычно называют Kanban, был впервые использован в 2006 году в компании Билла Гейтса Corbis. Она специализировалась на медиалицензиях. Эти практики быстро распространились в большом и многообразном международном сообществе, которое продолжает развивать и улучшать этот подход. Метод стали использовать как инструмент для определения и улучшения процессов, связанных с интеллектуальной активностью, в частности, как один из Agile-методов организации работ по разработке ПО.

Kanban стал символом визуализации рабочего процесса. Он использует вытягивающую систему и обязательное ограничение незавершённой работы, благодаря чему увеличивается прозрачность и предсказуемость процедур, а работа становится равномерной.

Kanban как гибкий метод разработки ПО

Kanban — стратегия оптимизации потока поставки ценности посредством процесса, который использует визуальную вытягивающую систему, имеющую ограничение незавершённой работы. Метод управления разработкой, реализующий принцип «Точно в срок» и способствующий равномерному распределению нагрузки между работниками. При таком подходе весь процесс разработки прозрачен для всех членов команды.

Метод идеален для организации и упорядочивания поточной работы, например сервиса поддержки. Сервис — это процесс, в котором задействованы один или несколько участников и который выполняет работу по созданию нематериальных продуктов. У каждого сервиса есть заказчик — сторона, формулирующая потребность или требования к сервису и принимающая выполненную работу. Kanban-метод направлен на определение и улучшение сервисов, которые создают ценность для заказчика. Структура сервиса изображается как ряд последовательных этапов, через которые проходит задача («рабочий элемент») слева направо.

Это наглядная система управления задачами, показывающая, что необходимо разрабатывать, когда и сколько. Задачи по мере поступления наносятся на карточки и крепятся на доску в виде списка к исполнению. Каждый разработчик может извлечь нужную задачу, взять её в работу и по мере выполнения двигать её по этапам процесса (например, «В работе», «Тестирование», «Завершено»), соблюдая определённые правила. Саму последовательность, набор этапов и правила перехода с одного этапа на другой определяет команда.

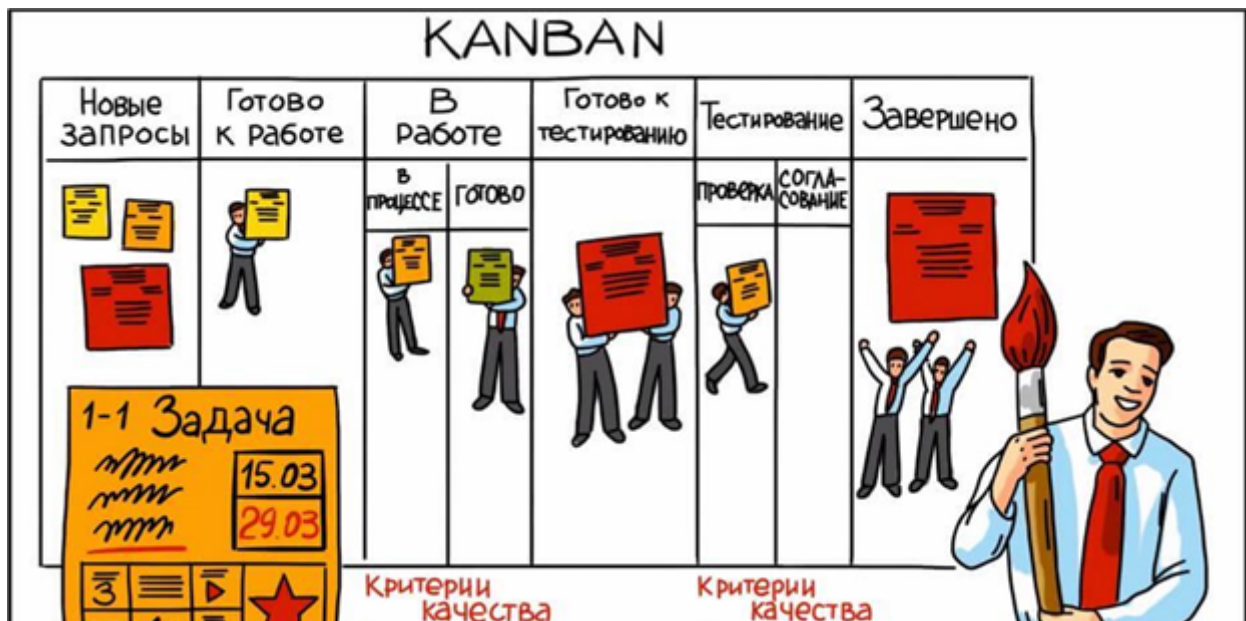


Рис. 6. Пример Kanban-доски

Чтобы такая поточная система стала Kanban-системой, должны выполняться обязательные условия и практики.

1. **Визуализация.** Необходимо визуализировать текущее состояние процессов. Kanban-доска — один из таких способов. Метод не предписывает, как должна выглядеть доска. Помимо столбцов вы можете включить в неё горизонтальные строки для различных типов задач (классы сервисов).
2. **Ограничение на количество незавершённой работы.** В системе должны присутствовать сигналы, ограничивающие количество незавершённой работы — WIP-лимиты (WiP – лимиты). Введение и соблюдение лимитов заменяет систему проталкивания задач на вытягивающую систему, в которой работа над новыми элементами начинается только по завершении (или выбрасывании) предыдущих задач. Подобные лимиты не позволяют скапливаться большому количеству частично завершённых работ. Если происходит блокировка потока работ в результате проблем с каким-либо элементом, это служит сигналом, по которому вся команда должна сосредоточиться на решении проблемы с целью восстановить поток.
3. **Определение точки принятия обязательств** — заказчик подтвердил желание получить рабочий элемент, а сервис подтвердил возможность выполнения этого элемента.
4. **Определение точки поставки** — принятие заказчиком выполненного рабочего элемента по определённым критериям, которые были зафиксированы в момент взятия обязательств.



Рис. 7. Kanban-доска: незавершённая работа

5. **Управление потоком.** Поток задач в Kanban-системе должен способствовать увеличению ценности, минимизировать время производства и повышать предсказуемость процесса. Необходимо постоянно контролировать процесс, выявлять узкие места (скопление задач на одном этапе) и заблокированные задачи (выполнение которых блокируют внешние факторы).
6. **Явные правила работы.** Чёткие и понятные правила позволяют определить и изложить процесс, выходящий за рамки определения жизненного цикла. Правила процессов должны отличаться минимализмом, простотой и наглядностью. Kanban-система должна содержать правила распределения задач, критерии готовности и перехода с этапа на этап, критерии выбора новых задач и использование классов сервисов. Правила могут различаться для различных этапов процесса.

7. **Внедрение циклов обратной связи.** Kanban предусматривает семь способов получения обратной связи, или каденций. Каденции — это регулярные совещания и проверки, приводящие к эволюционным изменениям и эффективному предоставлению сервисов. Важно выбрать оптимальную частоту каденций: слишком частые проверки могут привести к внесению новых изменений до того, как стал понятен эффект от предыдущих. А если проверки проводятся слишком редко, низкие показатели будут сохраняться слишком долго. Kanban предусматривает семь каденций (собраний): собрание по пополнению (внесение новых задач), Kanban-митинг, планирование поставки (запуска) и различные ревью (обзоры): сервиса, рисков, стратегии, операционное.
8. **Совместное улучшение, эволюция на основе экспериментов.** Kanban — метод совершенствования. При этом сам метод не предписывает конкретных действий, которые необходимо выполнить для улучшения вашего процесса. По результатам работы выдвигайте гипотезы о том, что можно улучшить в процессе, проверяйте свои предположения на практике и делайте выводы.

Все перечисленные практики дают способность видеть, в чём заключается рабочий процесс, а также вовлекают в совершенствование процессов, то есть сохранение и распространение полезных изменений, включая корректировку и сдерживание неэффективных изменений.

Принципы Kanban

Существует шесть основополагающих принципов, которые можно разделить на две группы.

Принципы управления изменениями

1. Начните с того, что есть сейчас: поймите текущий процесс и практики; соблюдайте текущие роли, обязанности и должностные инструкции.
2. Договоритесь между собой добиваться улучшений путём эволюционных изменений.
3. Поощряйте проявление лидерства на всех уровнях.

Эти три принципа призваны свести сопротивление переменам со стороны команды к минимуму и воспользоваться существующим опытом как базой для дальнейшего совершенствования эволюционным путём в процессе внедрения метода Kanban.

Принципы, связанные с организацией работ

4. Выясните потребности и ожидания заказчика, сосредоточьтесь на них.
5. Управляйте работой, а не людьми; дайте людям возможность организовать работу вокруг неё.
6. Развивайте правила для улучшения бизнес-показателей и увеличения пользовательской удовлетворённости.

При соблюдении этих принципов вся организация работ должна быть направлена на удовлетворение потребностей клиента и быструю поставку качественного результата.

Ценности Kanban

В основе метода лежат ценности, которые базируются на убеждении, что взаимное уважение всех лиц, делающих совместный вклад в общее дело, обеспечивает не только успех такого предприятия, но и его целесообразность. Выделяют следующие девять ценностей:

1. **Прозрачность** — открытый обмен информацией улучшает поток создания бизнес-ценности.
2. **Баланс** — в целях повышения эффективности необходимо обеспечить баланс между различными аспектами, точками зрения и возможностями.
3. **Сотрудничество** — совместная работа, совместное совершенствование.
4. **Клиентоориентированность** — конечная точка каждой Kanban-системы — создание ценности, то есть получение заказчиком требуемого продукта или сервиса.
5. **Поток** — работа представляет собой поток создания ценности. Способность видеть этот поток — ключевая отправная точка.
6. **Лидерство** — способность вдохновлять окружающих на действия своим примером, словами и идеями.
7. **Понимание** — знание (индивидуальное и организационное) о себе самом в целях дальнейшего развития, поскольку Kanban представляет собой метод совершенствования.
8. **Согласие** — обязательство совместного движения в сторону достижения целей с учётом (а если возможно, урегулированием) расхождений во мнениях и различий в подходах. Здесь речь идёт не о консенсусном управлении, а о принятии совместного стремления к совершенствованию.
9. **Уважение** — высокая оценка и понимание окружающих. Располагается в конце списка, поскольку это основа, на которой базируются остальные ценности.

Роли в Kanban

Kanban был и остаётся методом «начните с того, что есть сейчас», который не требует новых ролей, обязанностей и позиций в организации. Тем не менее на практике всё же появились две необязательные, но часто используемые роли. Теперь они определённым образом характеризуются в рамках метода. В данном случае важно не назначение на должность, а цель создания роли. Роль может выполнять отдельный специалист или член команды:

1. **Менеджер запросов для сервиса** (Service Request Manager, SRM) несёт ответственность за изучение потребностей и ожиданий заказчиков, содействие выбору и приоритизации рабочих элементов на собраниях по пополнению. Другие названия для роли: менеджер продукта, менеджер сервиса.
2. **Менеджер поставки для сервиса** (Service Delivery Manager, SDM) несёт ответственность за поток работы, в ходе которого выбранные рабочие элементы поставляются заказчику, за обеспечение проведения Kanban-митингов и собраний планирования поставки. Другие названия роли: менеджер потока, менеджер поставок и даже мастер потока.

Преимущества Kanban

Сейчас Kanban — один из популярных методов разработки, развития и поддержки ПО, используемый Agile-командами. Он предоставляет командам любых размеров ряд дополнительных преимуществ, касающихся планирования задач и обеспечения производительности. Помимо совершенной визуализации, гибкости планирования и сокращения времени цикла Kanban полезен тем, что ограничивает количество задач в работе. Чем больше задач, тем чаще приходится переключаться между ними, что сказывается на сроках их завершения, таким образом многозадачность убивает эффективность работ. Ограничение задач позволяет быстро находить в работе команды проблемные места, вызванные нехваткой внимания, людей или навыков.

Kanban-метод только в первом приближении кажется очень простым и понятным. На самом деле, помимо общих правил организации процесса, на практике он предполагает проведение достаточно большого количества математических расчётов для анализа эффективности работы: скорости поставки, времени производства, стоимости задержки и т. д. с использованием специальных формул, а также построение на основании полученных данных графиков и диаграмм. В этом курсе мы делаем только общий обзор данного метода. Детальную информацию о расчётах по методу Kanban можно найти в двух официальных документах: «Краткое руководство по Kanban» Дэвида Дж. Андерсона, Энди Кармайкла и «Руководство по Kanban для Scrum-команд» (разработано и поддержано Scrum.org, Даниелем Ваканти и Ювалом Йеретом).

Scrum

История создания Scrum

В 1986 году Хиротака Такеучи (Hirotaka Takeuchi) и Икуджиро Нонака (Ikujiro Nonaka) опубликовали в журнале Harvard Business Review статью The New New Product Development Game («Разработка нового продукта. Новые правила игры»), в которой изложили принципы командно-ориентированного подхода. Компании-лидеры рынка того времени (Xerox, Honda, Canon) вместо классической эстафеты по передаче продукта по этапам от одного функционального специалиста к другому применяли подход, похожий на игру в регби, когда команда продвигается по дистанции как единое целое, постоянно передавая мяч друг другу. Так зарождался Scrum.

В 1993 году Джефф Сазерленд (Jeffrey Sutherland) столкнулся с невыполнимой задачей: разработать менее чем за шесть месяцев новое ПО для одной известной IT-компании. Имея опыт работы с нестандартными методологиями, Джефф начал с того, что прочитал сотни статей о повышении производительности организации и поговорил с десятками ведущих менеджеров продукта. В ходе этого исследования его заинтересовали несколько оригинальных идей. Одна пришла из лаборатории Белла (той самой, где родился цикл Деминга): в одной статье о командной работе рассказывалось о пользе коротких ежедневных встреч команды и том, как это улучшает синхронизацию и повышает производительность команды. Но ключевая концепция была взята Сазерлендом из статьи Нонаки и Такеучи о «регбийном» подходе.

Изучив множество идей из различных источников, Сазерленд создал новый способ разработки ПО, дав ему название Scrum по аналогии с регби. Слово Scrum в переводе с английского означает «потасовка, драка за мяч, схватка вокруг мяча (в регби)». Scrum позволил Сазерленду успешно завершить практически невозможный проект в срок, в рамках бюджета и с небывало низким количеством багов. Вскоре Сазерленд объединился со своим коллегой Кеном Швабером (Ken Schwaber) для формализации подхода, и в 1995 году Scrum был впервые представлен общественности на IT-конференции OOPSLA.

Определение Scrum

Scrum — это процессный фреймворк, который начали использовать для управления работой над сложными продуктами в начале 90-х годов XX века. Это не техника или исчерпывающий метод, напротив, Scrum — фреймворк, в котором можно использовать разнообразные процессы и методы. Он выявляет несовершенства в управлении продуктом и методах работы, чтобы вы могли постоянно улучшать продукт, команду и рабочее окружение. Фреймворк помогает решать изменяющиеся в процессе работы задачи, чтобы поставлять клиентам продукты с максимально возможной ценностью. Все основные положения и принципы его использования содержатся в руководстве по Scrum, авторы которого Кен Швабер и Джефф Сазерленд. Руководство по Scrum содержит описание фреймворка и рассказывает о его ролях, событиях, артефактах и правилах.

Основные характеристики Scrum

1. Компактен и прост для понимания, но требует некоторых усилий для совершенного овладения.
2. Объединяет небольшую команду людей, которые сосредоточены на скорейшей поставке продукта, удовлетворяющего клиента.
3. Каждая отдельная команда чрезвычайно гибка и адаптивна. Эти преимущества распространяются на любое количество команд в организации: одну, несколько или целые сети команд, которые разрабатывают, выпускают, эксплуатируют и поддерживают продукты, таким образом объединяя труд тысяч людей. Они совместно работают и взаимодействуют благодаря продвинутым архитектурам и современным средам для выпуска.
4. Основан на теории эмпирического управления (эмпиризме). Согласно этой теории, источник знаний — опыт, а источник решений — реальные данные.
5. Использует итеративный подход (англ. iteration — «повторение») — выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы. При этом подходе продукт в каждой фазе развития проходит повторяющийся цикл PDCA: планирование — реализация — проверка — корректировка (англ. plan-do-check-act cycle).
6. Использует инкрементный подход (от англ. increment — «увеличение») — метод разработки программного обеспечения путём постепенного наращивания функционала, пока продукт не будет закончен. Продукт определяется как законченный, когда он удовлетворяет всем требованиям.

Элементы фреймворка и его организация

SCRUM



Рис. 8. Схема работы по Scrum

Основные элементы фреймворка: Scrum-команды и связанные с ними роли, события, артефакты и правила. Каждый элемент фреймворка служит определённой цели и обязателен для успешного использования Scrum.

Роли

1. **Product owner (владелец продукта)** — представитель со стороны заказчика.
2. **Scrum-master (скрам-мастер)** — роль, призванная помогать всем участникам процесса эффективно выполнять свою работу через понимание теории, практики, ценностей и правил Scrum.
3. **Development team (команда разработки)** — команда, состоящая из всех необходимых в проекте специалистов: бизнес-аналитик, архитектор, специалист по интеграции, дизайнер, фронтенд- и бэкенд-разработчики, тестировщики и т. д.

Артефакты (лат. artefactum от arte — «искусственно» + factus — «сделанный») — искусственно созданный объект, продукт человеческой деятельности. В рамках Scrum это:

1. **Product Backlog (бэклог продукта)** — перечень задач, поступивший от клиента. Эволюционирует вместе с постоянно меняющимися условиями проекта, чтобы продукт всегда оставался актуальным и конкурентоспособным.
2. **Sprint Backlog (бэклог спринта)** — перечень задач, взятых в спринт из бэклога продукта.
3. **Increment (инкремент)** — готовый функционал, который добавляет новую ценность продукту.

События

1. **Sprint (спринт)** — итерация длиной от 1 до 4 недель.
2. **Sprint planning meeting (планирование спринта)** — составление перечня задач (бэклога спринта), над которыми будет трудиться команда разработки во время спринта.
3. **Daily standup meeting (ежедневный Scrum)** — ежедневная встреча команды разработки (не более 15 минут), в ходе которой команда планирует свою работу на ближайшие 24 часа.
4. **Sprint review (обзор спринта)** проводится в конце спринта с целью демонстрации инкремента продукта стейкхолдерам, а также адаптации бэклога продукта.
5. **Sprint retrospective (ретроспектива спринта)** проводится после обзора спринта перед планированием следующего спринта с целью внутреннего аудита и создания плана внедрения улучшений для будущего спринта.
6. **Release (релиз)** — выпуск продукта, который может быть результатом всего проекта или происходить после нескольких спринтов или каждого спринта.

Правила Scrum связывают вместе события, роли и артефакты, регулируют отношения и взаимодействия между ними.

Описать процесс применения всех элементов фреймворка можно следующим образом: клиент по определённым правилам формирует бэклог продукта и определяет, что войдёт в спринт, команда оценивает работы, формирует бэклог спринта и приступает к реализации. В процессе реализации команда пользуется доской визуализации и ежедневно встречается, чтобы обсудить текущее положение дел. В конце спринта происходит демонстрация инкремента (готового функционала), ревью (обзор) спринта с участием клиента и ретроспектива внутри команды. Этот цикл повторяется раз за разом необходимое количество спринтов. Как только продукт или его часть, по мнению клиента (владельца продукта), будет готов к выпуску, произойдёт релиз, то есть запуск.

Основные принципы Scrum

Прозрачность

Значимые характеристики процесса и все его элементы должны быть известны и понятны всем участникам проекта, которые отвечают за его результат. Участники должны добиться согласия в понимании критериев оценки работ и их приёмки, оперировать общей терминологией, поддерживать соблюдение принятых всеми правил.

Инспекция

Участники процесса должны регулярно инспектировать артефакты и свой прогресс с помощью событий Scrum. Это позволяет вовремя обнаружить нежелательные отклонения и повысить качество работы и результата.

Адаптация

Если в результате инспекции будут обнаружены новые обстоятельства, которые влияют на достижение цели проекта и требуют изменений процесса или каких-либо условий, то эти изменения необходимо принять и адаптироваться к ним. Чем раньше будут внесены изменения, тем меньше риск дальнейших отклонений.

Ценности Scrum

Scrum-команда опирается в работе на следующие ценности:

1. **Преданность** команде, цели проекта, правилам Scrum, философии Agile.
2. **Смелость** брать на себя ответственность, действовать и работать над решением сложных задач.
3. **Сфокусированность** — удовлетворение потребностей клиента, фокус на цели спринта.
4. **Открытость** — обмен информацией и мнениями, несмотря на возникающие трудности.
5. **Уважение** к коллегам по команде как к самостоятельным личностям и профессионалам.

Такой подход создаёт атмосферу всеобщего доверия. Участники Scrum-команды исследуют и постигают эти ценности по мере работы с событиями, ролями и артефактами Scrum. Успешность использования Scrum напрямую зависит от того, насколько в команде придерживаются этих ценностей.

Преимущества Scrum

Сам по себе фреймворк Scrum прост. Понять правила, артефакты, мероприятия и роли несложно. Он задаёт структуру, при этом в ней есть свобода выбора — всё это исключает белые пятна в процессе разработки и позволяет в должной мере учесть специфику разных компаний. Сложные задания можно упорядочивать в легко выполнимые задачи, ориентированные на функционал, а значит, Scrum идеально подойдёт для сложных проектов. Благодаря тому, что роли и плановые мероприятия чётко разграничены, на протяжении всего цикла разработки сохраняются прозрачность и коллективная ответственность. Частый выпуск продуктов мотивирует команду и гарантирует удовлетворённость пользователей, ведь они видят, как продукт развивается в течение короткого отрезка времени.

Применение Scrum

С начала девяностых этот фреймворк активно используется по всему миру в условиях резкого роста сложности технологий и ежедневно подтверждает свою пользу и необходимость применения. Scrum доказал свою особую эффективность в итеративной и инкрементальной передаче знаний.

При разработке программного обеспечения он применяется для следующих целей:

1. Исследовать и выявлять жизнеспособные рынки, технологии и возможности продуктов.
2. Разрабатывать продукты и улучшать их.
3. Выпускать продукты и их обновления по несколько раз в день.
4. Разрабатывать и поддерживать облачные технологии (онлайн, безопасно, по требованию) и другие среды для использования продуктов.
5. Поддерживать и обновлять продукты.

Также Scrum широко используется в работе над любыми другими продуктами и услугами, в управлении организациями.

За свою историю Scrum применялся для разработки оборудования, автономных транспортных средств и микробиологических исследований. Использовался в школах, медиакомпаниях, правительстве, в том числе ФБР, маркетинге и управлении организациями — повсеместно и повседневно в жизни отдельных людей и сообществ.

Сравнение Scrum и Kanban

Kanban и Scrum — это инструменты процесса, которые помогают работать эффективнее и до определённой степени подсказывают, что делать.

Суть Kanban: начать с того, что есть сейчас, визуализировать работы, ограничить объём незавершённой работы (WiP), регулировать поток работ в процессе реализации и, как итог, достичь максимальной эффективности (или скорости). Kanban-команды стремятся максимально сократить время, которое уходит на выполнение проекта от начала до конца. Для этого они используют Kanban-доску и непрерывно совершенствуют рабочий процесс.

Задача Scrum-команд — поставить работающее ПО за ряд промежутков времени (спринтов). Scrum-команды используют особые роли, создают специальные артефакты и проводят регулярные собрания, чтобы работа шла в нужном русле и поддерживался командный дух.

На первый взгляд это два абсолютно разных инструмента, у которых мало общего, однако при ближайшем рассмотрении можно увидеть очевидные сходства.

Сходства:

- придерживаются общих принципов и основных ценностей Agile-концепции;
- используют вытягивающие системы планирования;
- ограничивают объём работ;
- ориентированы на ранние и частые поставки продукта;
- используют прозрачность и визуализацию для улучшения процесса;
- план релиза постоянно оптимизируется на основе эмпирических данных.

Несмотря на базовые сходства, инструменты используют разные подходы к организации процесса, контрольным точкам, ролям и управлению изменениями.

Различия

Критерий сравнения	Scrum	Kanban
Организация работ	Обязательное использование всех элементов Scrum	Начинаем с того, что есть
График	Регулярные спринты фиксированной длительности (например, две недели)	Непрерывный процесс. Могут быть отдельные ритмы для планирования, выпуска и усовершенствования процессов. Также могут быть событийно-управляемые итерации вместо ограниченных по времени
Объём работы на момент времени	Объём задач ограничен и предопределён, для их выполнения отводится фиксированный промежуток времени — спринт	Ограничен установленным количеством незавершённой работы
Оценка задач, приоритизация работ и планирование	Обязательно	Опционально
Подходы к релизу	В конце каждого или нескольких спринтов после одобрения владельцем продукта	Поставка выполняется непрерывно или на усмотрение команды
Роли	Владелец продукта, Scrum-мастер, команда разработчиков	Обязательных ролей нет, но могут быть введены менеджер запросов и менеджер поставки
Ключевые показатели	Скорость (производительность)	Время выполнения, время цикла, объём незавершённой работы (WiP)
Отношение к изменениям	В ходе спринта команды стремятся избегать изменений: изменения приведут к неверным выводам, касающимся оценки задач	Можно менять приоритеты и содержание заданий. Изменение может произойти в любой момент, ограничение лишь одно — количество незавершённой работы (WiP)
Доска визуализации	Обнуляется и обновляется после каждого спринта, принадлежит одной команде	Используется на протяжении всего жизненного цикла проекта, может принадлежать нескольким командам, например тестирование и релиз осуществляются отдельной группой

Исходя из сравнения, можно сделать три важных вывода:

1. И Scrum, и Kanban — высокоадаптивные инструменты, но Scrum более директивен, чем Kanban. Scrum предоставляет вам больше правил, которым необходимо следовать, и ограничений. Таким образом, он оставляет меньше вариантов для выбора. Kanban оставляет почти всё на

усмотрение команды. Единственные ограничения: визуализировать ваш ход работы и ограничить незавершённую работу (WiP).

2. И Scrum, и Kanban можно назвать эмпирическими в том смысле, что ожидается, что вы будете экспериментировать с процессом и заточите его под свои условия. Ни Scrum, ни Kanban не содержат ответов на все вопросы: они просто дают вам базовый набор ограничений, чтобы было от чего отталкиваться при улучшении вашего процесса.
3. Kanban не может включать в себя Scrum, а Scrum может включать в себя элементы Kanban как дополнительные техники для совершенствования процесса.

Именно на основе третьего вывода некоторые команды объединяют идеалы Scrum и Kanban в Scrumban. Из Scrum они берут полноценный фреймворк или большую его часть, а из Kanban, например, концентрацию на правилах и ограничение количества одновременно выполняемых задач.

Scrumban

Считается, что Scrum лучше подходит для продуктов и развивающихся проектов, а Kanban — для поддержки и производства. По этой причине в последнее время всё более популярным становится Scrumban, который сочетает в себе лучшие функции обоих подходов.

В Scrum вы заранее выбираете работу, которую будете выполнять в следующем спринте. У вас есть временные рамки одного спринта, чтобы закончить отобранные задачи.

В Kanban ограничение задаёт только параметр WiP. Это означает, что вы можете изменить элементы в очереди в любое время, а конца спринта просто нет. Работа течёт непрерывно.

Scrumban = Scrum + Kanban

Чтобы внедрить Scrumban, вам потребуется полностью овладеть и Scrum, и Kanban. Поэтому неверно думать, что это более лёгкий фреймворк.

В Scrumban используются доска для визуализации и ограничение на работу в процессе. Элементы из бэклога спринта выполняются последовательно, в колонку «В работе» не попадают сразу все карточки (хотя это не противоречит Scrum Guide). Поток работы выравнивается. Команды смотрят на процесс перехода карточек по доске, чтобы увидеть возможности для совершенствования. Kanban работает внутри спринта, ограничивая число незавершённых задач в колонке «В работе».

В Scrumban мы можем выполнять планирование итераций через регулярные промежутки времени, синхронизированные с обзором и ретроспективой спринта. При этом цель планирования — заполнить не всю очередь задач заранее, а лишь часть. Это значительно упрощает планирование. Сэкономленное время тратится на лучшую подготовку задач.

Вот как может быть организован процесс по Scrumban. Роли участников не обязательны, поэтому, например, отсутствует роль Scrum-мастера. Владелец или менеджер продукта формирует бэклог продукта, расставляет приоритеты. Команда определяет длительность регулярных итераций, количество

этапов в потоке, которые будут проходить задачи, проставляет ограничение на количество незаконченных задач и определяет правила, по которым задачи переходят с одного этапа на другой. Бэклог спринта отсутствует, команда оценивает несколько самых приоритетных задач из списка и приступает к работе, далее по мере выполнения вытягивает из бэклога новые задачи. Ежедневные встречи, обзор спринта и ретроспектива не отменяются.

В данном случае Kanban, внедрённый во фреймворк Scrum, позволит команде ускорить цикл работы над задачами, повысить качество за счёт дополнительных правил и избежать авралов за счёт ограничения количества незавершённых задач.

Приведённый выше алгоритм — лишь один из примеров реализации проекта по методу Scrumban. Он может корректироваться в сторону Scrum или Kanban в зависимости от того, что нужно проекту и команде. Чаще всего гибрид этих двух техник подходит для следующих проектов:

- проекты технического обслуживания;
- событийная работа;
- справочная служба или информационная поддержка;
- удалённые команды, которые работают над запуском новых продуктов;
- работа, предшествующая разработке, исследования;
- тестирование и развёртывание продукта;
- компании, где есть проблемы с чистым Scrum.

Заключение

На втором уроке курса мы рассмотрели самые популярные Agile-подходы к организации процесса разработки. Дали определение, основные характеристики, принципы и ценности метода Kanban и фреймворка Scrum. Сравнили эти инструменты и пришли к выводу, что их гибрид может повысить скорость и качество работы.

На практике мы будем использовать Scrum: он лучше всего подходит для цели нашего курса — разработки нового продукта. Следующий урок будет посвящён подробному рассмотрению ролей и их обязанностей в Scrum, а также принципу их взаимодействия на проекте.

Глоссарий

Артефакты (лат. artefactum от arte — «искусственно» + factus — «сделанный») — искусственно созданный объект, продукт человеческой деятельности.

Инкрементальность — приращение результатов предыдущего этапа.

Итеративность — регулярное повторение полного цикла работы над продуктом с непрерывным анализом результатов предыдущего этапа и корректировкой требований и процесса.

Каденции (обратная связь) — регулярные совещания и проверки, приводящие к эволюционным изменениям и эффективному предоставлению сервисов.

Фреймворк (англ. framework — «каркас, структура, рамки, основа») — набор базовых элементов и правил, своего рода каркас, на котором строится процесс разработки.

Daily standup meeting (ежедневный Scrum) — ежедневная встреча команды разработки (не более 15 минут), в ходе которой команда планирует свою работу на ближайшие 24 часа.

Development team (команда разработки) — команда, состоящая из всех необходимых в проекте специалистов: бизнес-аналитик, архитектор, специалист по интеграции, дизайнер, фронтенд- и бэкенд-разработчики, тестировщик и т. д.

Increment (инкремент) — готовый функционал, который добавляет новую ценность продукту.

Kanban-доска — инструмент управления Agile-проектами, с помощью которого можно наглядно представить задачи, ограничить объём незавершённой работы и добиться максимальной производительности или скорости.

Product Backlog (бэклог продукта) — перечень задач, поступивший от клиента. Эволюционирует вместе с постоянно меняющимися условиями проекта, чтобы продукт всегда оставался актуальным и конкурентоспособным.

Product owner (владелец продукта) — представитель со стороны заказчика.

Release (релиз) — выпуск продукта, который может быть результатом всего проекта или происходить после нескольких или каждого спринта.

Scrum-master (скрам-мастер) — роль, призванная помогать всем участникам процесса эффективно выполнять свою работу через понимание теории, практики, ценностей и правил Scrum.

Sprint (спринт) — итерация длиной от 1 до 4 недель.

Sprint Backlog (бэклог спринта) — перечень задач, взятых в спринт из бэклога продукта.

Sprint planning meeting (планирование спринта) — составление перечня задач (бэклога спринта), над которыми будет трудиться команда разработки во время спринта.

Sprint retrospective (ретроспектива спринта) проводится после обзора спринта перед планированием следующего спринта с целью внутреннего аудита и создания плана внедрения улучшений для будущего спринта.

Sprint review (обзор спринта) проводится в конце спринта с целью демонстрации инкремента продукта стейкхолдерам, а также адаптации бэклога продукта.

WiP-лимит (Work in Progress limit) — ограничение количества выполняющейся в данный момент работы (WiP). Ограничения WiP повышают пропускную способность команды и сокращают объем почти выполненной работы, заставляя команду сосредоточиться на меньшем наборе задач.

Дополнительные материалы

1. [Всё о Kanban за 4 минуты \(видео\)](#).
2. [Что такое Kanban-доска \(статья\)](#).
3. [Kanban WiP limits — Agile Coach \(видео на англ. яз.\)](#).
4. [Всё о Scrum за 4 минуты \(видео\)](#).
5. [Подробно об основных принципах фреймворка Scrum \(видео\)](#).
6. [Год в Scrum: наблюдения Scrum-мастера / Блог компании Росбанк / Хабр](#).
7. [Сравнение Kanban и Scrum \(статья\)](#).

Используемые источники

1. [Руководство по Scrum \(Кен Швабер, Джефф Сазерленд\)](#).
2. [Kanban. Краткое руководство \(Дэвид Дж. Андерсон, Энди Кармайкл\)](#).
3. [Руководство по Kanban для Scrum-команд \(разработано и поддержано Scrum.org, Даниелем Ваканти и Ювалом Йеретом\)](#).
4. [Scrum и Kanban: выжимаем максимум \(Хенрик Книберг и Маттиас Скарин\)](#).
5. [Что такое Kanban \(статья\)](#).
6. [Не Скрамом единым: где и как применять Канбан-метод \(статья\)](#).
7. [Что такое ScrumBAN?](#)
8. [13th Annual State of Agile Report \(2019\)](#).
9. [14th Annual State of Agile Report \(2020\)](#).

Практическое задание

1. Пройти [тестирование к уроку 2](#).
2. Ознакомиться с методическим пособием.