

Инструментальные средства информационных систем

Группа: БИН-19-1

Автор: Сафиуллина Н.Ф.

Лекция 5 Инструменты управления проектами и отслеживания выполнения задач

Содержание

Введение	2
1 Методики и методы управления проектом и их инструменты	3
1.1 Классический метод	3
1.2 Метод Scrum	5
1.3 Методы Agile	8
1.4 Метод Lean	11
1.5 Метод Kanban	12
1.6 Выбор метода управления проектом	14
2 Инструменты управления проектами	16
2.1 Диаграмма Ганта	16
2.2 Битрикс24	19
2.3 Trello	20
2.4 Teams	21
2.5 Burndown chart	22
2.6 Рекомендуемая литература	30

Введение

Рассмотрим некоторые инструменты для управления проектами и отслеживания выполнения задач. Прежде чем переходить непосредственно к рассмотрению инструментов необходимо познакомиться с теоретической базой. Поэтому в этой лекции изучим следующее:

- основные понятия проектной деятельности,
- методики управления проектами (Waterfall, Scrum, Agile, Lean, Kanban),
- подходы применяемые в проектной деятельности (Gantt chart, Kanban-доска),
- измерение трудоемкости задач,
- цели, которые все методики и подходы преследуют.

Что такое проект?

Проект – это временное предприятие или некая временная организованная команда, предназначенные для создания продукта, услуг или достижения каких-то результатов.

Управление проектом – это особый вид деятельности, базирующийся на предварительной коллегиальной разработке комплексной модели действий, в результате которых будет достигнута некоторая оригинальная цель, и направленный (*вид деятельности направленный*) на реализацию этой модели.

У предприятия обычно два вида деятельности: проектная и операционная. Проектная деятельность отличается от операционной. Операционная деятельность тоже состоит из задач, но главное отличие: у проекта есть временные рамки.

Проект имеет следующие **характеристики**:

1. Проект имеет дату начала и дату окончания.
2. У проекта есть результат, выраженный в измеряемых показателях.
3. Проект направлен на достижение этого результата.
4. У проекта есть команда проекта.
5. У проекта есть жизненный цикл.
6. У проекта есть миссия, т.е. некоторая генеральная цель, это причина существования проекта.

Об управлении проектами всегда говорят с точки зрения проектного менеджера, и рассматриваются инструменты для управления проектами для менеджера, но ваша задача (как начинающих специалистов) сначала научиться управляться своим пулом задач и научиться планировать время. Конечно, начинающий специалист сначала будет исполнителем в проектах.

Цель изучения этого материала – понять как отвечать на главный вопрос заказчика к исполнителю проекта.

Можете предположить какой главный вопрос заказчика или менеджера проекта к исполнителю проекта?

Это: “Когда?” “Когда будет сделана задача?” “Когда вы возьметесь за задачу?”

Обычно заказчики хотят все и сразу. Вот пример обсуждения будущего задания на разработку ПО. “... и если кладовщик при приеме товара отсканировал товар с коротким сроком годности, то пусть ТСД, ручной терминал-сканер, сделает свето-звуковую индикацию...”, “Простите, что сделает?”, “Свето-звуковую индикацию, т.е. запищит и замигает”, “А, т.е. вы хотите, чтобы терминал вел себя как карета скорой помощи, а простое сообщение на экран вас не устроит?” “Да, не устроит! И ещё пусть завибрирует...(и тут же) Когда сделаете?”.

Очень сложно ответить на этот вопрос сразу. И ещё сложнее, если вы ни разу не видели этот терминал и не знаете, легко ли сделать так, чтобы он вибрировал, воспроизводил звуки и светился. А может, это вообще невозможно.

Но существует правильный общий ответ.

Ответ примерно такой: попрошу вас предоставить описание задач (даже еще не техническое задание), затем мы с командой оценим трудоемкость задач и оценим приблизительное время требующееся на разработку с учетом текущей загрузки.

Далее получаете задачи, дробите их на подзадачи, считаете трудозатраты и функциональные единицы (мы поговорим о них позже) и, с учетом средней скорости реализации функциональной единицы, рисуете графики и отвечаете на вопросы “когда?”. Если задача не слишком сложная, то можно сделать это в уме, прикинуть примерно. А с накоплением опыта оценка в уме постепенно превращается в интуицию.

1 Методики и методы управления проектом и их инструменты

1.1 Классический метод

Классический проект также называется: традиционный или каскадный или waterfall (т.е. водопад). Называется он так потому что в таких проектах одна фаза идет за другой последовательно и, фазы не повторяются, вода не течет обратно по водопаду. Состоит из типовых этапов:

1. инициация
2. планирование
3. реализация
4. мониторинг и контроль
5. завершение



Этап инициации.

На этапе инициации происходит:

- описание проблемной области,
- постановка цели,
- описание миссии,
- формирование команды,
- определение заинтересованных сторон,
- формирование матрицы ответственности.

Для формулирование цели часто пользуются аббревиатурой SMART:

S – specific – цель должна быть конкретной,

M – measurable – цель должна быть измеримой,

A – achievable – цель должна быть достижимой,

R – relevant – цель должна быть обоснованной,

T – time bound – должно существовать ограничение во времени.

Например:

1. “Повысить продажи свежих молочных продуктов” – нехорошая цель, не правильно сформулированная.
2. правильно примерно так: “к концу первого квартала 2023 года, повысить долю продаж молочной продукции местных производителей с 7% до 10% в целях поддержать локальные фермерские хозяйства”.

NB. Формулировка задач должна быть максимально простая, состоящая из глаголов и существительных, т.е. сделать что-то.

Этап планирования.

Нас больше интересует этап планирования, так как на этапе планирования применяется множество различных инструментальных средств. На этапе планирования:

- описывают задачи,
- указывают сроки решения задач,
- описывают риски и бюджет.

Так как каскадный метод планирования подразумевает выполнение задач одна за другой, то применяются подходы календарно-сетевого планирования, например диаграммы Ганта (Генри Гантт), вернёмся к ней позже.

Этап реализации.

На этапе реализации выполняются запланированные работы.

Мониторинг и контроль.

Это непрерывная деятельность в течении проекта, которая включает анализ текущего процесса и промежуточных результатов и корректировку хода проекта.

Этап завершения.

Проект считается завершен, если:

- его результаты удовлетворяют установленным критериям на результаты,
- составлена документация,
- прошло обучение стороны заказчика, как пользоваться продуктом или услугой,
- конечный продукт передан заказчику.

Традиционный метод управления проектами считается устаревшим. Он не подходит для разработки программного обеспечения, потому что считается не реальной ситуация – знать все требования на входе в IT проект. Традиционный метод планирования всё так же применяется при строительстве, например, электростанций и прочих похожих проектах.

1.2 Метод Scrum

Scrum чаще всего используется для разработки программного обеспечения и в других IT проектах. Эта методика вышла из Apple и Google.

Scrum полезен в сложных сферах, где существует неопределенность и быстрая изменчивость.

Существуют описанные ценности scrum, например:

- фокусировка – фокус на нескольких задачах в спринт,
- смелость – реальная оценка проблем, не закрывая глаза на причины,
- ответственность – все ответственные за результат спринта,
- уважение – все участники должны уважать друг друга,
- открытость – открыто обсуждать в команде все ситуации, лучше переделать что-то на этапе разработки, чем копить технический долг.

Сам процесс работы над проектом подчиняется следующим принципам:

1. прозрачность,
2. инспекция,
3. адаптивность.

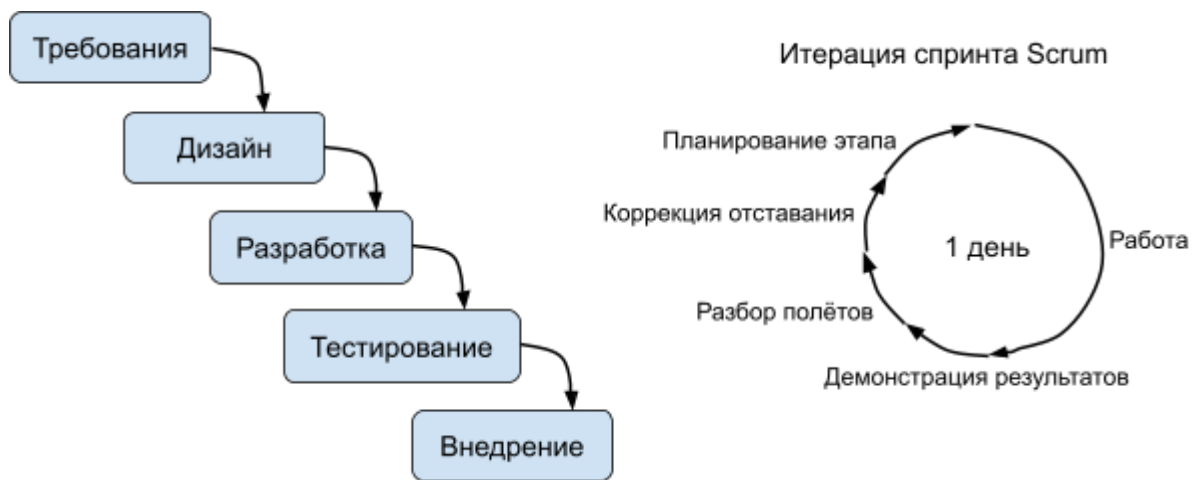
Scrum является итерационным методом управления проектом. Процесс работы в Scrum цикличен. Большой отрезок времени, который потребуется для решения поставленной задачи, разбивают на короткие отрезки, которые называют спринты. Максимальная длина одного спринта примерно один месяц, но обычно спринт равен 1-2 недели. Каждый спринт состоит из ежедневных scrum.



Давайте сравним характеристики двух методов.

	Waterfall	Scrum
1	Все требования известны в начале и не меняются.	Требования могут меняться по ходу проекта.
2	Объемное и подробное ТЗ.	Список “историй” пользователей, backlog.
3	Тестирование в конце.	Тестирование после каждой итерации.
4	Негибкая система.	Гибкая система.
5	Готовый продукт получают в конце.	Работающую часть продукта (часто называют “рыба”) получают после каждой итерации
6	Клиент не видит промежуточных результатов.	Клиент видит и может влиять на промежуточный результат.

Сравним схемы двух методик:



Кроме спринта, к scrum относятся следующие понятия:

- бэклог (backlog) продукта,
- бэклог (backlog) спринта,
- пользовательская история,
- ежедневный scrum,
- scrum доска.

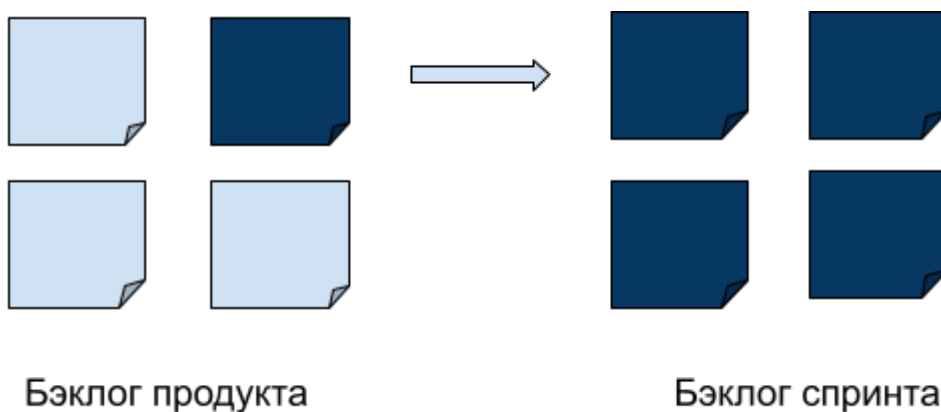
Бэклог продукта – это функциональные требования к продукту, описание что должен уметь делать продукт, какие возможности предоставлять или какой он должен быть, т.е. это эквивалент технического задания классического метода управления проектом.

Бэклог спринта – это часть требований, взятых из бэклога продукта, которые будут реализованы в текущем спринте.

Оба бэклога состоят из пользовательских историй.

Пользовательская история – это краткое и понятное описание того, какая функция или свойство требуется продукту.

Ежедневный scrum – ежедневная ретроспективная встреча участников scrum команды, примерно 15 минут и обычно утром, на которой обсуждается, что сделано вчера, что будет сделано сегодня, что помешало достигнуть результатов вчера.



Scrum доска – настоящая или виртуальная доска, где пользовательские истории перемещаются по следующим этапам работы над ними: stories (ожидает), to do (надо делать), in progress (в процессе), testing (требуется проверка), done (выполнена) по согласованию с заказчиком.

Stories	To do	In progress	Testing	Done
задача →	→	→	→	→ задача

1.3 Методы Agile

Agile – набор методов и практик для гибкого управления проектами в различных прикладных областях. Это тоже итерационный метод.

Существуют ценности agile, изложенные в Agile manifesto:

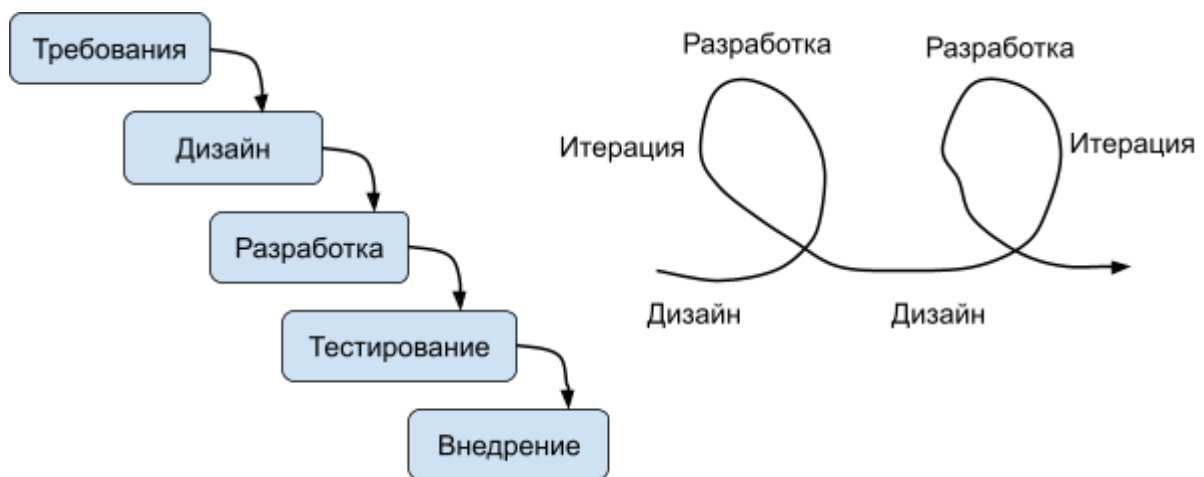
1. Продукт важнее документации.
2. Взаимодействие с заказчиком важнее контракта.
3. Способность к быстрым изменениям важнее намеченного плана.
4. То, что вы делаете, важнее должности и регалий.

Там же, в Agile manifesto, изложены 12 принципов:

1. Работающий конкурентоспособный продукт, удовлетворяющий заказчика – лучший показатель прогресса и измеритель эффективности.
2. Оперативная и бесперебойная поставка продукта, удовлетворяющая заказчика.

3. Адаптивность продукта к новым требованиям, которые могут повысить его ценность и конкурентоспособность, возможность внесения изменений на любом этапе разработки.
4. Простота и прозрачность технических решений, документации, процессов и инструментов, чтобы не создавать лишней работы.
5. Частая поставка функционирующего продукта.
6. Постоянный темп работы всех участников проекта на протяжении всего его срока.
7. Минимизация организационных и информационных барьеров, лучший путь передачи информации – личная встреча.
8. Тесное и ежедневное общение исполнителей с заказчиком в течении всего проекта.
9. Мотивация команды и обеспечение всеми необходимыми условиями работы, поддержкой и доверием.
10. Самоорганизация и самоконтроль команды.
11. Непрерывное улучшение профессиональных компетенций команды проекта.
12. Систематический анализ и постоянный поиск возможностей оптимизации командной и индивидуальной работы.

Проведем сравнение waterfall и agile:



Каждая итерация это совокупность этапов:

1. планирование
2. проектирование
3. создание прототипа
4. тестирование
5. обратная связь
6. запуск

Где пятый этап может переходить либо в проектирование либо в запуск.



Преимущества agile:

1. быстрота,
2. адаптивность,
3. фокус на результат.

Недостатки agile:

1. Отсутствие документации, agile не отрицает документацию, просто она не в фокусе.
2. Краткосрочное планирование не всегда учитывает масштабируемость.
3. Появление новых требований после нескольких итераций может привести к изменению изначального бэклога.

1.4 Метод Lean

Строго говоря lean это не методология управления проектом, а философия, впрочем как и agile. Подход lean часто применяется в бережливом производстве. Фокусируется на сокращении времени процесса (например, создании программного кода), на снижении затрат и прочее. Таким образом lean направлен на борьбу с неэффективностью и потерями.

При производстве информационных систем могут быть следующие скрытые потери:

– интеллект – это выполнение работы не соответствующей уровню подготовки сотрудника;

- движение – лишние перемещения сотрудников, в поисках инструкций, исходных данных, согласования доработок;
- переделывания – исправление ошибок, команда должна следить за качеством кода, проводить регулярные ревью кода на этапе разработки и не тратить время на переделывания;
- ожидание – ожидание получения доступа к информационным ресурсам, ожидание поступления задач, и прочее;
- недоделанная работа – это может недописанная функциональность, которую решили сделать по-другому или задачу передали другому исполнителю, код остается брошен в проекте и может мешать находить актуальный код;
- лишняя функциональность – функциональность добавленная в продукт, но не используемая потребителями;
- потеря ответственности – при передачи задачи от одного исполнителя к другому и так далее, может произойти потери ответственного за разработку;
- переключение – когда исполнитель делает одновременно несколько задач или проектов, при переключении между ними требуется время, чтобы вспомнить контекст.

Существует 10 принципов бережливого производства, думаю, для IT сферы подойдут следующие:

1. Ликвидировать мусор – удалить из процесса лишнее, удалить код, который является мусором.
2. Минимизировать запасы – не создавать лишнюю функциональность.
3. Максимизировать поток – устранить ожидания.
4. Производство зависит от потребительского спроса – создавать только нужную функциональность.
5. Знать требования клиентов – ориентироваться на требования клиента и пользователей, если клиент говорит, что так будет удобно, даже если разработчики так не считают, следовать требованиям клиента.
6. Сделать правильно с первого раза – этот принцип легко сказать, чем сделать, но главное не лениться, даже если чтобы сделать правильно потребуется больше сил и времени, лучше сделать правильно, без “костылей”.
7. Расширять возможности работников – организовывать обучение новому и развитие навыков сотрудников.
8. Построить систему с легкой заменой ее деталей – к разработке информационных систем это можно соотнести так: создавать универсальный продукт, блочно-модульную систему, где одну часть можно легко заменить не повредив общую работоспособность, этот пункт немного противоположен п.2, нужно найти ту грань между универсальностью и излишеством.

9. Создать культуру постоянного совершенствования.

Недостатки lean подхода:

- не каждая часть проекта требует достаточного внимания, т.е. невозможно понять достаточно ли была проработана некоторая часть, все ли необходимые действия были сделаны;
- в отличие от Scrum не предполагает деления работы на кусочки или итерации, что может привести к растягиванию сроков проекта.

В любом случае ценность Lean в том, чтобы создавать продукт и минимизировать при этом потери.

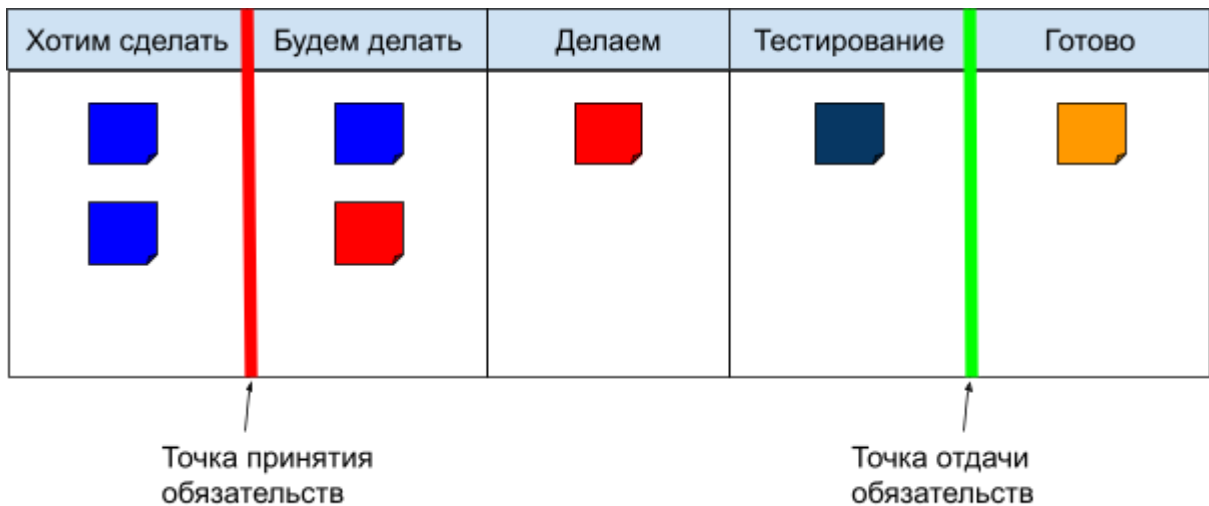
1.5 Метод Kanban

Kanban – это набор практик и принципов, т.е. по существу мы не можем назвать его чисто методом управления проектами, название образовано от японского “карточка” или “сигнал”, метод основывается на движении карточек по этапам, управлять проектом и контролировать процесс можно благодаря визуализации движения этих карточек на канбан-доске.

Kanban также является итерационным подходом, итерации называются петлями обратной связи или каденциями.

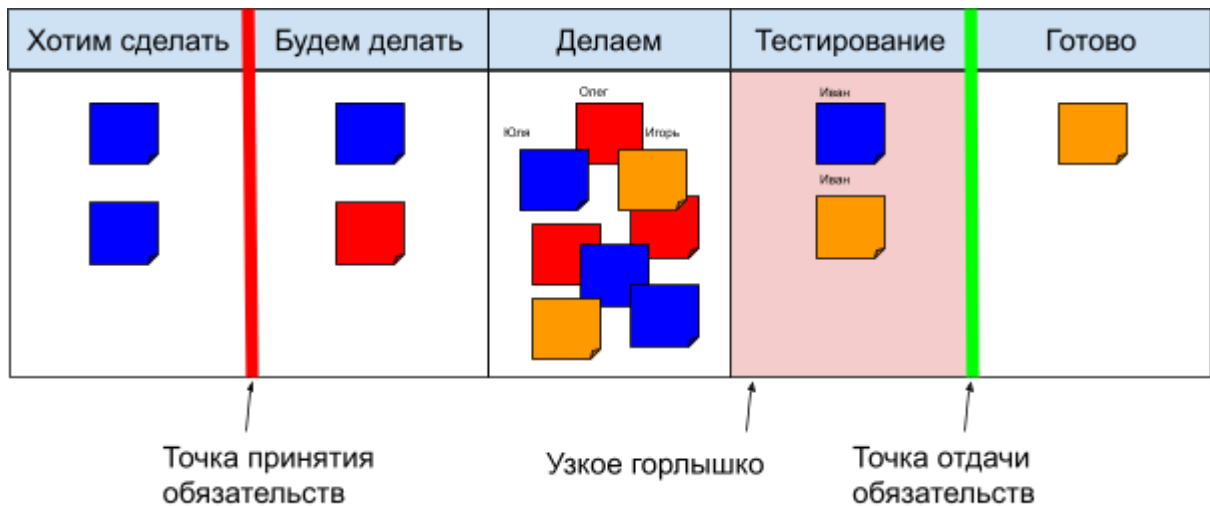
Основные понятия, относящиеся к Kanban:

1. Карточка – для каждой задачи существует своя карточка, похожа на пользовательскую историю из scrum.
2. Ограничение на количество карточек – на каждом этапе должно быть не более 5-6 задач.
3. Поток операций Waterflow – задачи выполняются постоянно в порядке приоритета.
4. Визуализация – движение карточке отображается на канбан-доске.
5. Каденции - петли обратной связи.
6. Канбан-митинг – ежедневная встреча, где обсуждают проблемы застрявших задач, кроме этой встречи существует еще несколько организационных встреч, например, по постановке задач.



Что мы видим на доске Kanban? Мы видим:

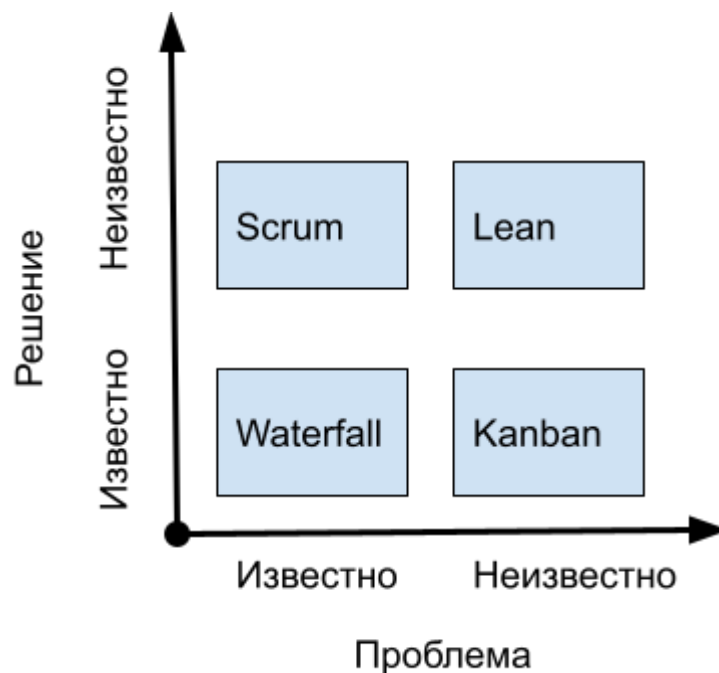
1. Колонки – это этапы, которые проходят карточки, в отличие от scrum колонки соответствуют рабочим этапам процесса.
2. Карточки разных цветов – разные цвета привязаны к различным типам задач.
3. Исполнитель – к каждой карточке привязан исполнитель, который занимается карточкой в данный момент.
4. Движение карточки – карточка движется слева направо, от колонки “Инициализация” к колонке “Готово”.
5. Точка принятия обязательств (commitment point) – это разделитель между колонками, задачи находящиеся в колонках левее этого разделителя находятся на стадии рассмотрения “делать задачу или нет”, а задачи в колонках правее – приняты к реализации.
6. Точка снятия обязательств – это разделитель между колонками, обычно находится перед колонкой “Готово”, когда карточка переходит её, значит задача готова и ее отдают заказчику.
7. “Узкое горлышко” – это тот этап, перед которым скапливаются карточки, значит на этом этапе требуется разгрузка, требуется разобраться откуда проблема, почему карточки медленно проходят этот этап.



Kanban подходит для проектов без жесткого deadline. Помогает отслеживать процесс, минимизировать количество незавершенной работы и выявлять проблемы в рабочем процессе.

1.6 Выбор метода управления проектом

Выбор метода управления проектом зависит от того насколько понятна или не понятна сама проблема и, насколько известно или неизвестно решение.



Если проблема известна, значит есть четкое описание цели, остается понять известно ли какие шаги помогут решить поставленную задачу. Если шаги

известны, то можно применить классический подход (waterfall). Если на входе еще не известно, что придется делать, чтобы решить задачу (типичная IT задача), то нужно выбрать Scrum.

Если особой цели достичь чего-то нет (например, тех поддержка или сопровождение) или нет строгого deadline, но известно, что надо делать, известны задачи и подзадачи, то применяем Kanban.

Если что-то надо менять (например, есть какие-то потери и неэффективность работы), что-то идет не так, и при этом непонятно что и как менять, то можно пойти по пути бережливого производства (Lean).

По сути именно методиками являются только scrum и waterfall, методики имеют определенный набор инструментов, применяя которые получают результат. При отказе от использования части инструментов методики, это уже не будет выбранная методика.

Agile, lean и kanban можно назвать методами и подходами (даже философией), которые имеют общую рекомендацию в каком направлении двигаться, чтобы получить результат, можно применять все практики этих методов, а можно применять только часть из них.

2 Инструменты управления проектами

Далее рассмотрим инструменты, подходящие для автоматизации описанных методик управления проектом. Вот некоторые из них:

1. Microsoft Project
2. GanttPRO
3. Ganttic
4. Битрикс24
5. Trello
6. Teams
7. Burndown chart

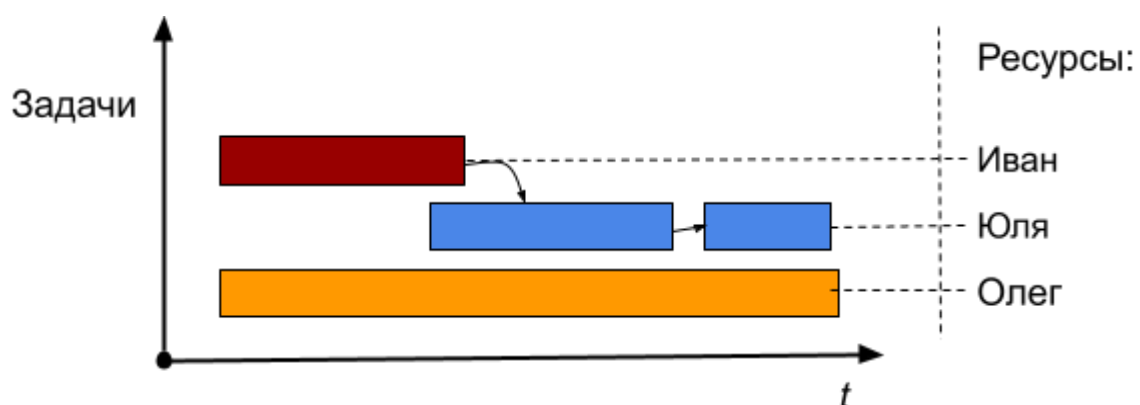
Про MS Project отдельно говорить не будем, это достаточно популярный легкодоступный инструмент, немного устаревший, но легкий в освоении, похож на все продукты MS Office и входящий в него. Позволяет строить диаграммы Ганта, создавать задачи и анализировать процесс выполнения.

Отдельно рассмотрим “Диаграмму сгорания задач” (Burndown chart), который является частью фреймворка scrum.

2.1 Диаграмма Ганта

Первый инструмент, который мы упоминали в традиционном подходе диаграммы Ганта.

Диаграмма Ганта обычно состоит из двух частей или двух осей, на вертикальной оси располагаются задачи, а горизонтальная ось – это временная шкала, где гистограммы обозначают работу над задачами, работа имеет начало и окончание.



На диаграмме можно видеть:

- задачи,
- начало и окончание задачи,
- исполнителя,
- зависимости между задачами.

Диаграммы Ганта применяются в том случае, когда пул задач известен заранее, т.е. когда существует детально описанное техническое задание, когда строго определено какая задача следует за какой, а какие можно выполнять параллельно, когда существуют строгие дедлайны.

В итерационных методах управления проектами тоже часто применяется диаграмма Ганта, как способ спланировать отрезок работ.

С помощью диаграмм Ганта можно спланировать работы не только по проекту, но и составить план переезда, ремонта квартиры или организации мероприятия.

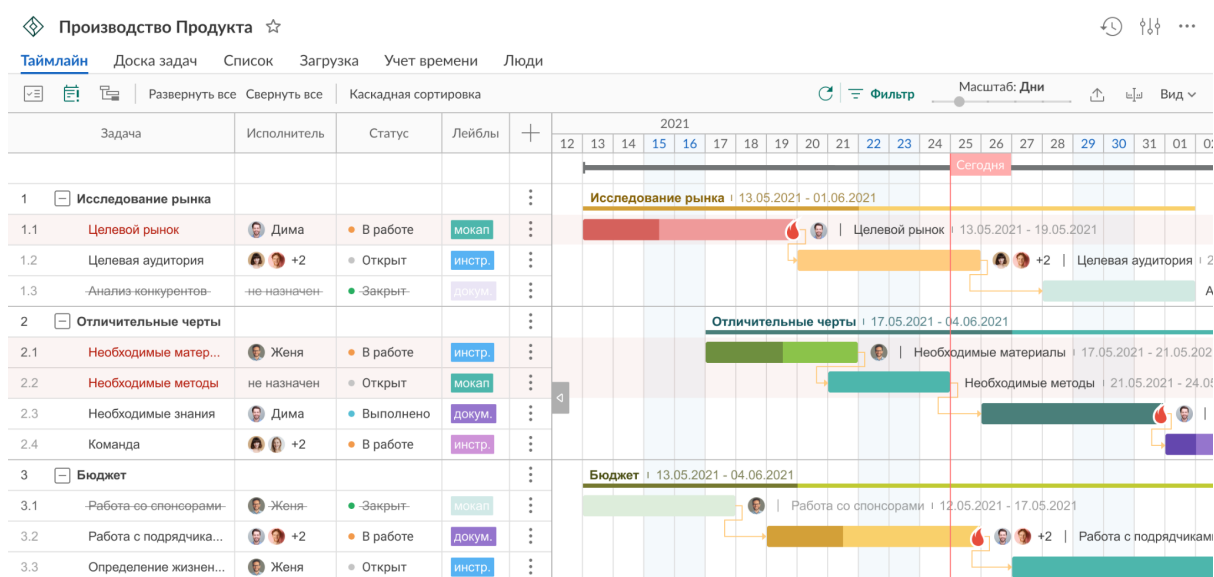
Диаграмму Ганта можно построить в MS Excel, это применяется повсеместно, но есть и специализированные инструменты, а часто диаграмма Ганта входит в крупные инструменты управления проектами, например:

- MS Project,
- Битрикс24,

Специализированные инструменты:

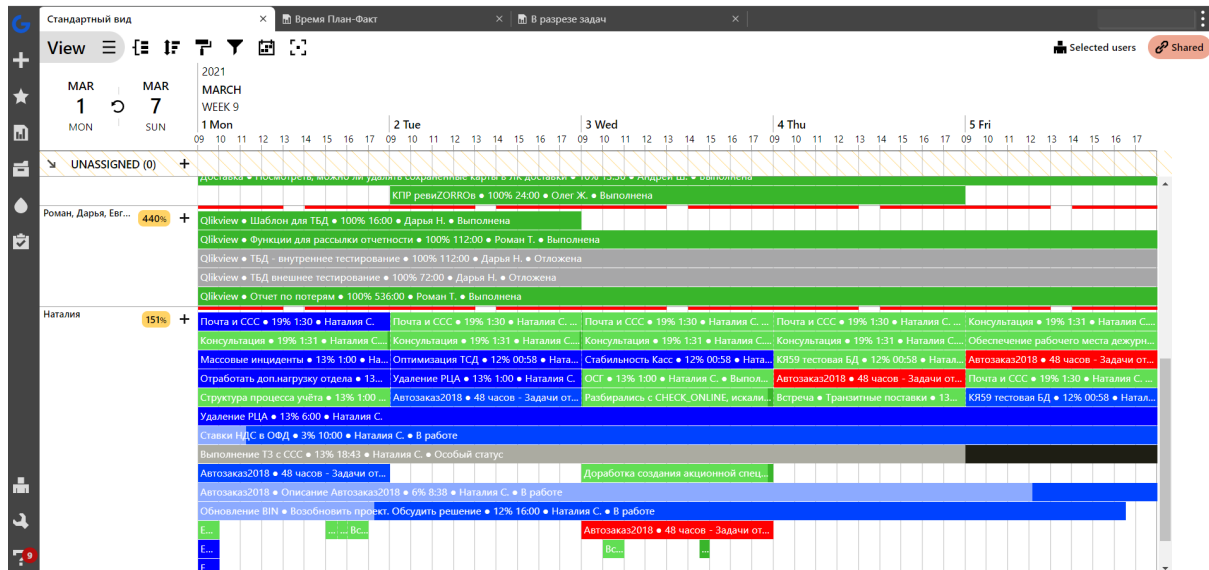
- GanttPRO.
- Ganttic.

Далее вид GanttPRO. Что мы тут видим? Мы видим три крупные задачи, которые разложены на подзадачи, видим исполнителей и статус выполнения задач. Определена текущая дата и мы видим, что некоторые задачи уже закрыты, а некоторые отстают от графика. Видно, какие задачи могут пересекаться во времени.



GanttPRO позволяет вычислять нагрузку и учитывать рабочее время.

Мы долгое время использовали веб-приложение Gantt.com им можно пользоваться бесплатно для команды до 10 человек. В данной интерпретации диаграммы Гантта слева мы видим исполнителей, а справа располагаются задачи исполнителя разграниченные по времени и занимающие свое место (день недели, несколько дней, месяцы) на шкале времени.



Кроме этого мы видим:

- статус задачи, от этого зависит цвет гистограммы;
 - видно сколько фактически потрачено времени на задачу, есть встроенный таймер, чтобы его запустить, нужно зайти в конкретную задачу и запустить таймер, когда работа над задачей закончится, нужно остановить таймер, потраченное время запишется в диаграмму;
 - процент, который задача должна занимать от общего рабочего времени.
- Что позволяет Gantt.com:
- можно просто вести учет рабочего времени, т.е. вносить задачи по факту выполнения, по таким данным в будущем можно создавать отчеты, видеть что занимает больше времени и прочее;
 - можно планировать выполнение задач, распределять время на задачи задавая распределение в процентах или часах;
 - можно создавать отчеты, например сравнивать запланированное время и время фактическое уделенное задаче, и другие отчеты;
 - можно вычислять дату завершения задачи, если известно сколько примерно часов она займет и сколько времени готовы уделять ей ежедневно или еженедельно.

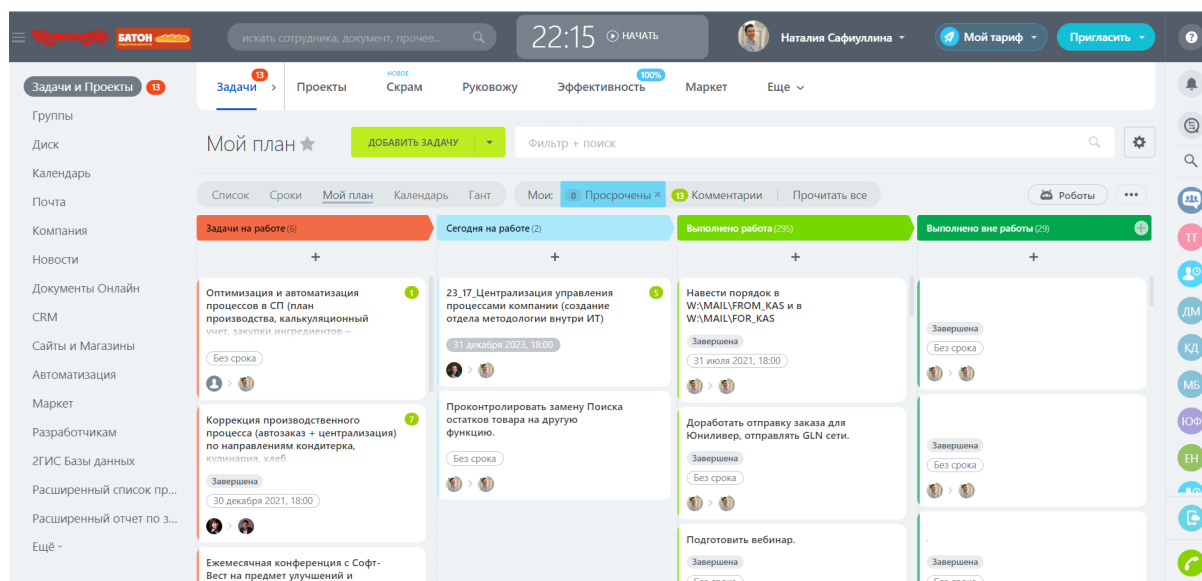
Самое главное, приложение позволяет прикидывать сроки и отвечать на вопрос “когда”.

2.2 Битрикс24

Битрикс24 – это российский продукт, позволяет сделать многое, у него есть и инструменты организации совместной работы (почта, чаты, календари и т.д), есть и CRM (управление отношениями с клиентами), есть и различные возможности автоматизировать работу компании. С помощью Битрикс24 даже можно создать сайт. Но нас интересует управление проектами и задачами.

Как пишут на самом сайте Битрикс24 для управления задачами и проектами мы можем пользоваться:

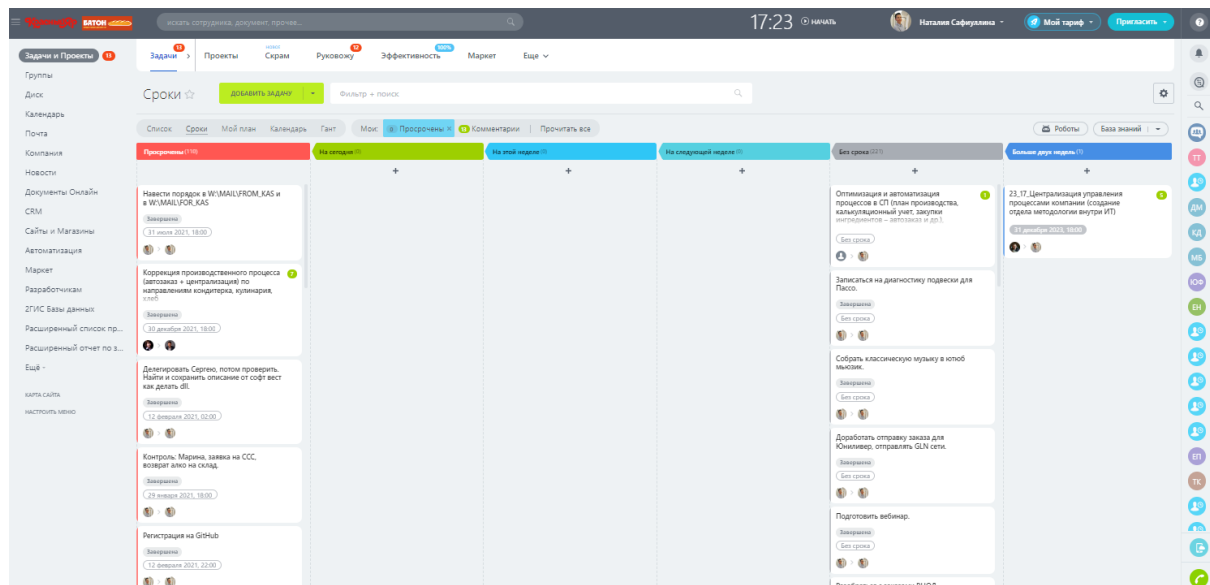
- диаграммой Гантта,
- доской Канбан,
- учетом времени,
- обсуждением в задачах,
- scrum (новинка, доступен в расширенной версии).



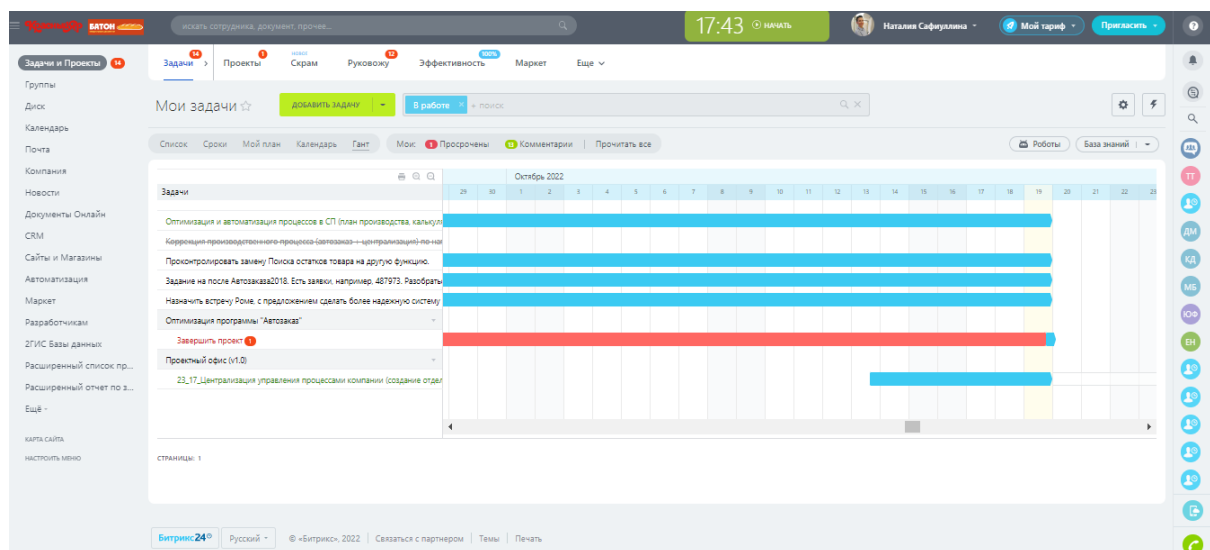
В разделе Задачи видим все задачи, один и тот же список задач можно просматривать в различных представлениях, и пользоваться любым методом отслеживания задач, какой удобнее.

Например, предыдущая картинка – это доска, на которой карточки задач передвигаются по этапам. Этапы и их названия можно задавать самостоятельно, т.е. доску “Мой план” можно превратить в полноценную Kanban-доску.

Следующая доска показывает распределение задач по времени, можно запланировать на ближайшее время.



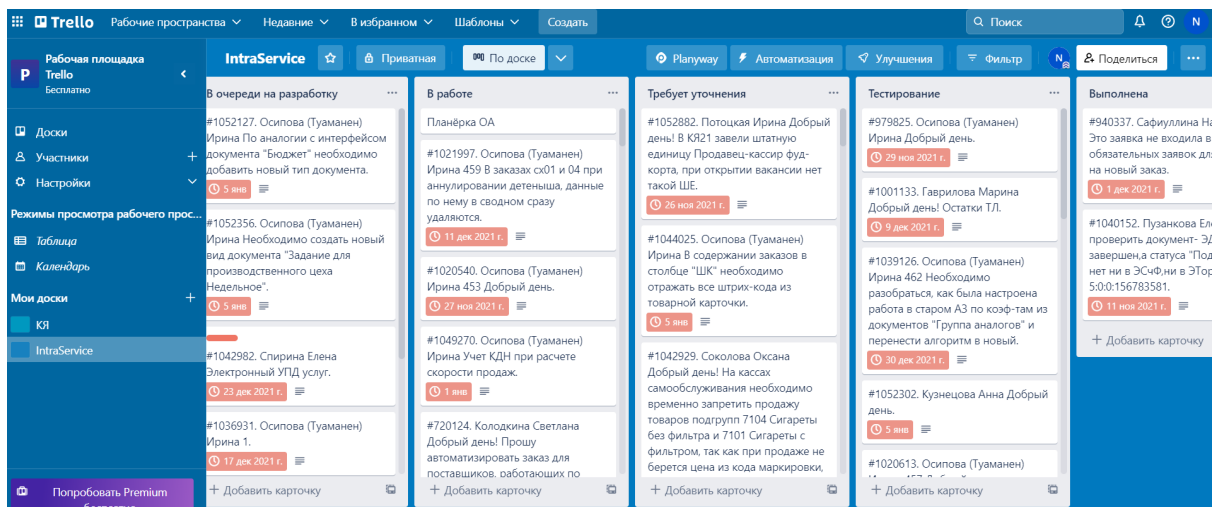
Битрикс24 также предоставляет возможность работать с задачи в виде Гантт-диаграммы. Слева на диаграмме задачи и подзадачи, справа гистограммы работы.



В Битрикс24 есть раздел проекты, где представлены такие же инструменты для работы с задачами проекта, как и в разделе задач.

2.3 Trello

Trello является визуализацией Kanban-доски, имеет веб-приложение и десктопную версию. Базовым приложением можно пользоваться бесплатно на сайте trello.com.



В данном приложении можно:

- создавать несколько тематических досок,
- создавать карточки задач,
- создавать карточки участников,
- обмениваться файлами,
- оставлять комментарии.

Это простое приложение, у него нет предустановленных аналитических возможностей, т.е. невозможно построить отчеты или вести учет рабочего времени.

Одним из плюсов Trello является наличие API. Например, можно связать Trello и Service Desk приложение (Service Desk – это поддержка), т.е. с приложением, где пользователи размещают свои истории, заявки на доработки, заявки на обслуживание, если service desk не имеют удобного инструмента отслеживания выполнения задач.

2.4 Teams

Teams продукт компании Microsoft и входит в Office 365, это платформа для совместной работы, включает:

- чаты,
- хранилище файлов,

- онлайн встречи,
- возможность вести заметки.

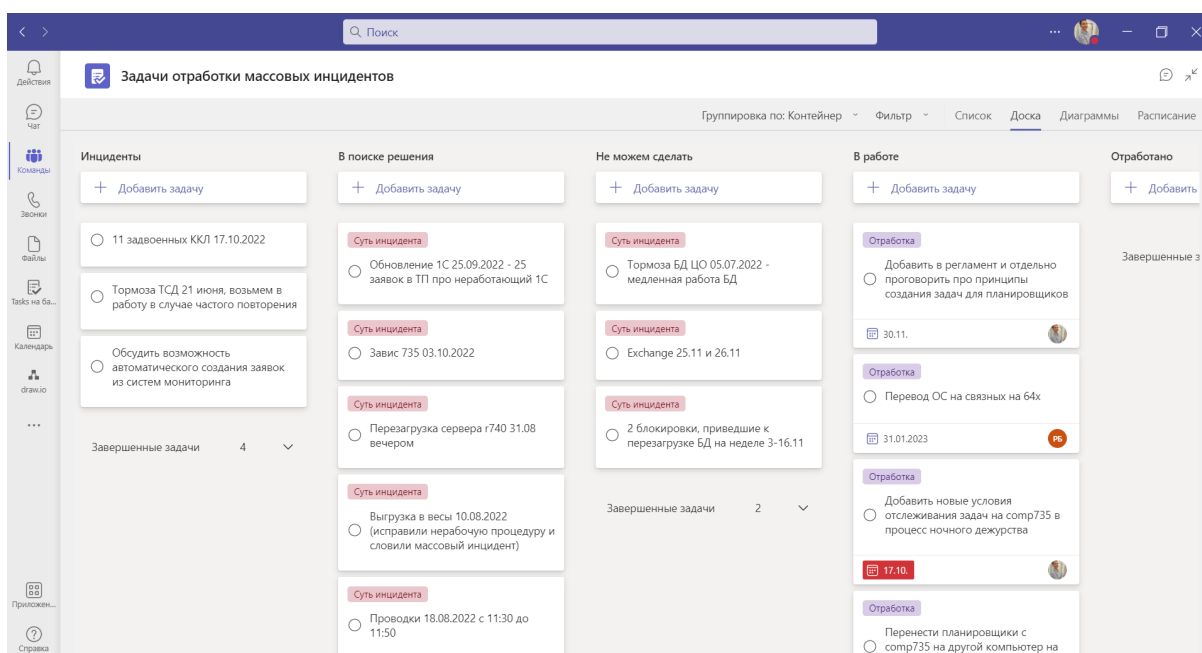
Более того, помимо основного функционала, в Teams можно устанавливать необходимые приложения.

Как применять teams для управления проектами? В teams можно создавать команды, внутри команд можно создавать отдельные каналы.

Канал соотносится одному проекту. В каждом канале по умолчанию включены:

- хранилище файлов,
- чат.

Для полноценного управления проекта нужно установить необходимые приложения в канал, например, приложение для создания доски задач.



В канале можно вести документацию проекта, можно создавать иерархию папок в хранилище файлов и описание проекта вести в Word и Excel документах. Или можно вести документацию в встроенном Wiki.

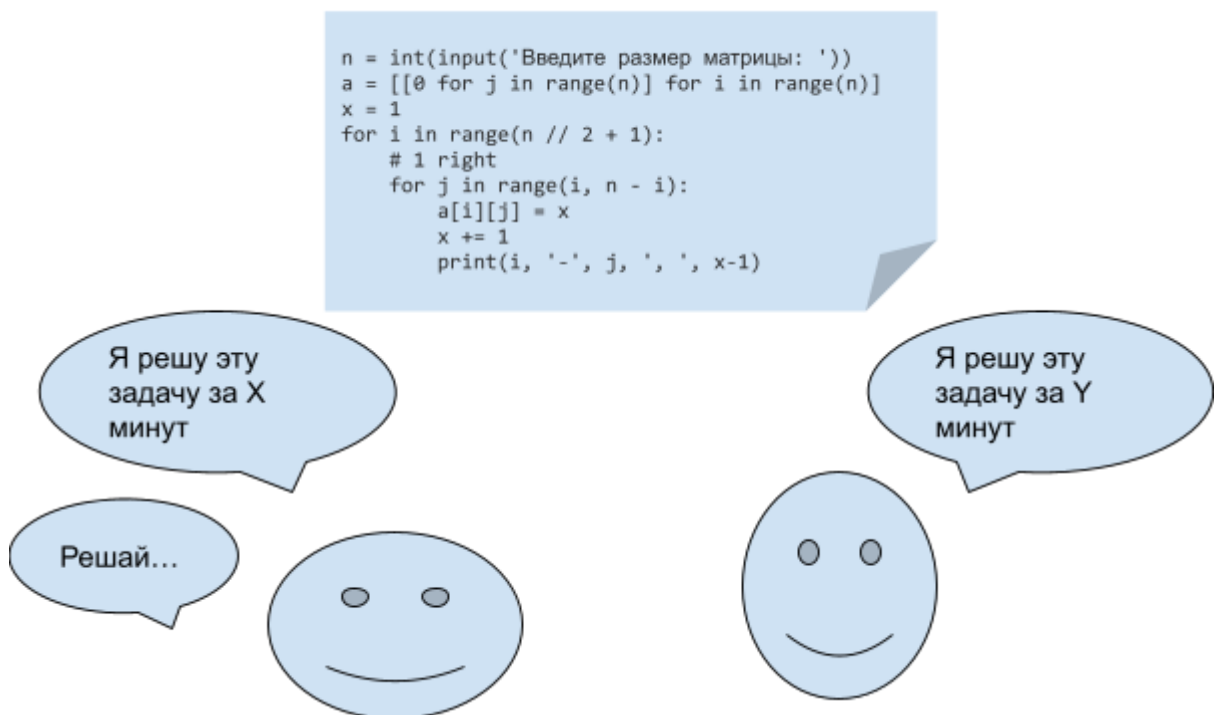
Интересно приложение Task на базе Планировщика и To do, тем что имеет мобильную версию и управлять задачами можно с телефона.

2.5 Burndown chart

Простой инструмент отслеживания хода работ можно сделать самостоятельно. Давайте построим Диаграмму сгорания задач в таблице, например MS Excel.

Диаграмма Burndown показывает объем работы, выполненной в спринте, а также общую оставшуюся работу. Диаграммы используются для прогнозирования вероятности того, что команда завершит свою работу к указанному сроку. Они также подходят для того, чтобы отслеживать увеличение объема работы.

Прежде чем начать строить диаграмму, сначала нужно понять, что не все задачи равны. Вес или объем задач можно оценивать в часах, сколько часов потребуется на решение той или иной задачи, но это не совсем удобно. Исполнители тоже не равны, исполнители будут тратить разное время на одну и ту же задачу. Поэтому нужна более универсальная оценка. Одним из методов оценки задач являются функциональные единицы.



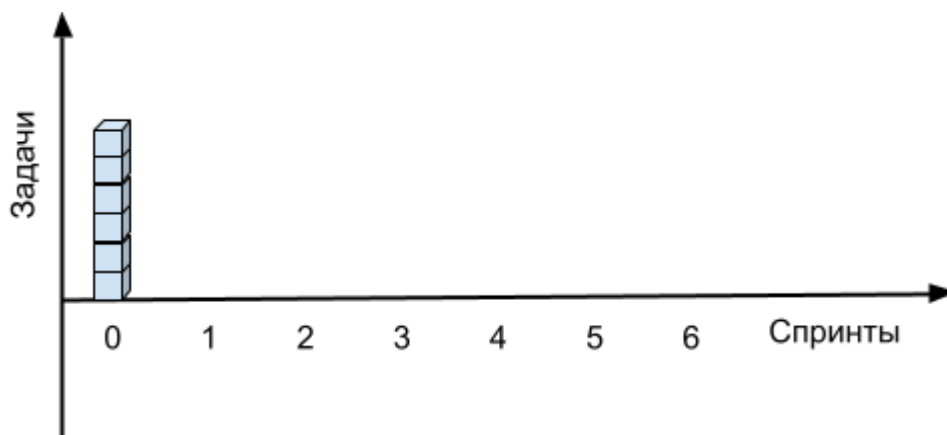
Функциональная единица программы – это кусочек программы, которая выполняет одно простое законченное действие. Более точно, что будет являться функциональной единицей, следует договориться с командой.

Итак, существует некоторый список задач, так как мы рассматриваем методику scrum, то задачи изначально известные поступают в нулевой спринт. Задачи оценены в размере функциональных единиц.

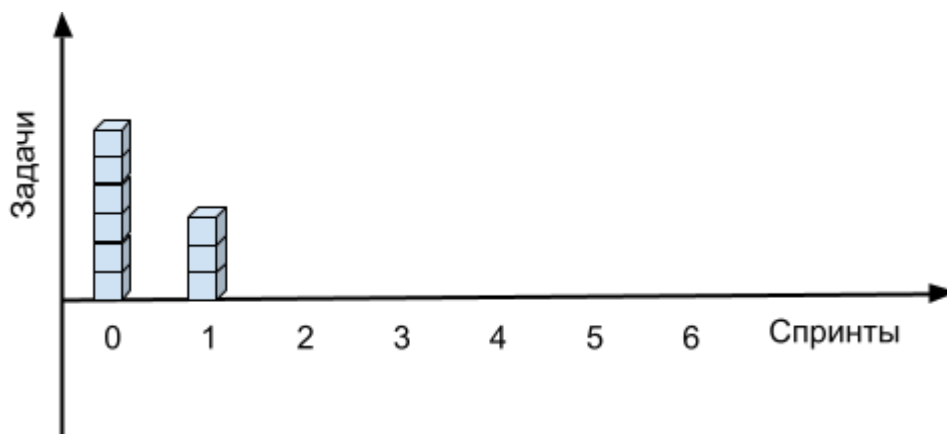
Пусть у нас есть 4 задачи:

1. Скачать Docker = 1 фе
2. Выяснить что с виртуализацией, включить ее = 2 фе.
3. Установить Docker = 1 фе.
4. Скачать и развернуть первый контейнер = 2 фе.

Приступим к созданию диаграммы, пусть задачи имеющиеся в какой-то спринт будут представлены в виде столбика гистограммы:



В первый спринт сделали 2 задачи, это 3 фе.



В этот же первый спринт выяснилось, что установка Docker не получается, программа не запускается. Добавляется новая задача, выяснить что не так, это прочитать и разобрать ошибки и найти решение в туториалах.

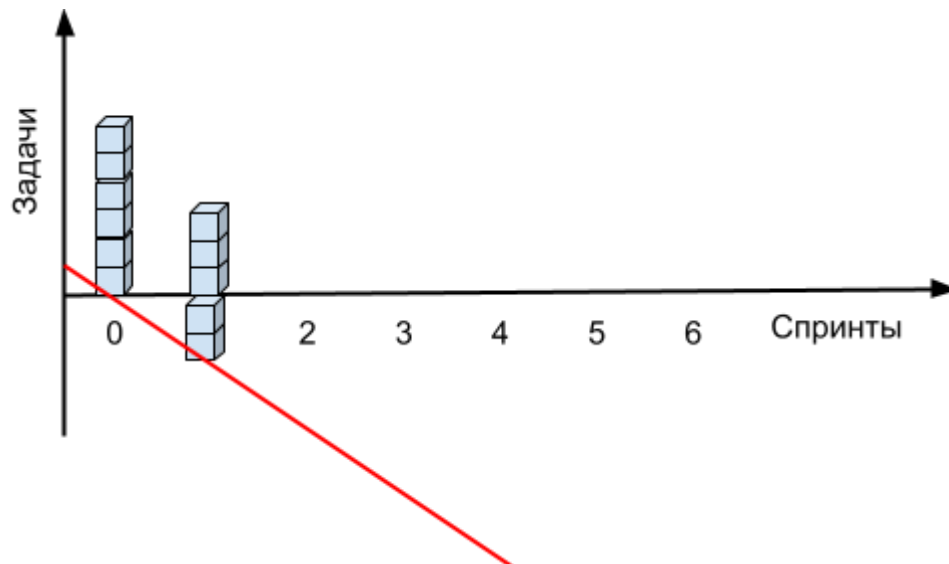
Новая задача:

5. Найти ошибку в запуске Docker, устранить ее = 2 фе.

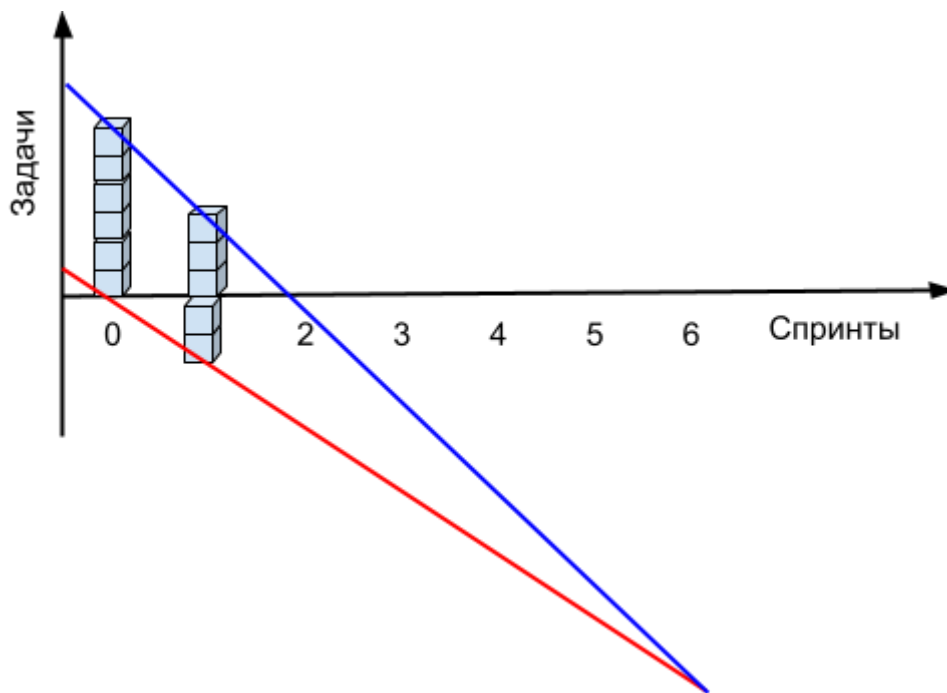
Что должно произойти с диаграммой? Как нарисовать новые задачи? Новые задачи добавляются ниже оси, т.е. к существующим задачам снизу.



Новые задачи добавляются снизу для того, чтобы нарисовать прямую которая показывает тенденцию добавления задач.



Дорисуем линию, которая отражает тенденцию выполнения задач.



Если скорость выполнения задач выше, чем скорость добавления задач, то линия пересекутся. Эта точка пересечения является тем спринтом, где на горизонте замаячит окончания всех работ. На рисунке в районе шестого спринта. Проверим:

Спринт 0 – пришло 6 фе.

Спринт 1 – решено 3 фе, пришло 2 фе, осталось 5 фе.

Если дальше так же пойдёт, то

Спринт 2 – решено 3 фе, пришло 2 фе, осталось 4 фе.

Спринт 3 – решено 3 фе, пришло 2 фе, осталось 3 фе.

Спринт 4 – решено 3 фе, пришло 2 фе, осталось 2 фе.

Спринт 5 – решено 3 фе, пришло 2 фе, осталось 1 фе.

Очевидно, что в шестой принт задачи кончатся. Если в разные спринты скорость добавления и решения задач различные, вычисляется среднее. Строится аппроксимирующая прямая.

Чтобы построить такую диаграмму в Excel, нужны три сущности:

1. backlog – таблица со списком задач с оценками трудности,
2. вычисления – промежуточные значения для построения диаграммы,
3. диаграмма.

Бэклог содержит как минимум следующие важные колонки:

1. описание задачи,
2. оценка задачи,

3. в какой спринт задача пришла,
4. в какой спринт задача ушла.

Задача	Оценка	Пришла	Ушла
Скачать Docker	1	0	1
Выяснить что с виртуализацией, включить ее	2	0	1
Установить Docker	1	0	
Скачать и развернуть первый контейнер	2	0	
Найти ошибку в запуске Docker, устранить ее	2	1	

Вычисления данных будут следующие:

Исходное количество задач = СУММЕСЛИ(Пришла; 0; Оценка) = 6
 – т.е. суммировать оценки задачи, если задача пришла в нулевой спринт, т.е. исходный список задач.

Последний спринт = МАКС(Ушла) = 1
 – т.е. если указан номер спринта, в который сделана какая-то задача, то мы берём максимальный указанный спринт и считаем, что это последний отработанный спринт.

Средняя скорость выполнения = СРЗНАЧ(Ушла) = 3

Средняя скорость поступления = СРЗНАЧ(Пришла) = 2

Осталось работы всего = Исходное количество задач
 - СУММ(Выполнено) + СУММ(Поступило) =
 5

Таблица вычислений имеет следующие колонки:

Поступило = ЕСЛИ(Спринт<=Последний спринт;
 СУММЕСЛИ(Пришла; Спринт; Оценка);"")

Выполнено = ЕСЛИ(Спринт<=Последний спринт;
СУММЕСЛИ(Ушла; Спринт; Оценка);"")

Осталось = ЕСЛИ(Спринт<=Последний спринт;
Осталось на предыдущем спринте - Выполнено; "")

Добавлено = ЕСЛИ(Выполнено<>"";
-Поступило + Добавлено на предыдущем спринте;
"")

Спринт	Поступило	Выполнено	Осталось	Добавлено
1	2	3	3	-2
2				
3				
4				

Диаграмма визуализирующая данные в Excel строится на основании типа диаграммы График.

На вертикальной оси располагаются значения из колонок Осталось и Добавлено, на горизонтальной – откладываем номер спринта.

Все пустые значения нужно скрыть, т.е. показывать пустую ячейку, если нет значения.

Нужно добавить следующие элементы диаграмм:

- линии тренда (линейный прогноз),
- полосы повышения и понижения.

Следует выбрать размер вертикальной оси такой, чтобы было удобно видеть данные.

И наконец основные линии графика можно скрыть и оставить только линии тренда и полосы повышения/понижения.

Пример заполнения данных в Excel.

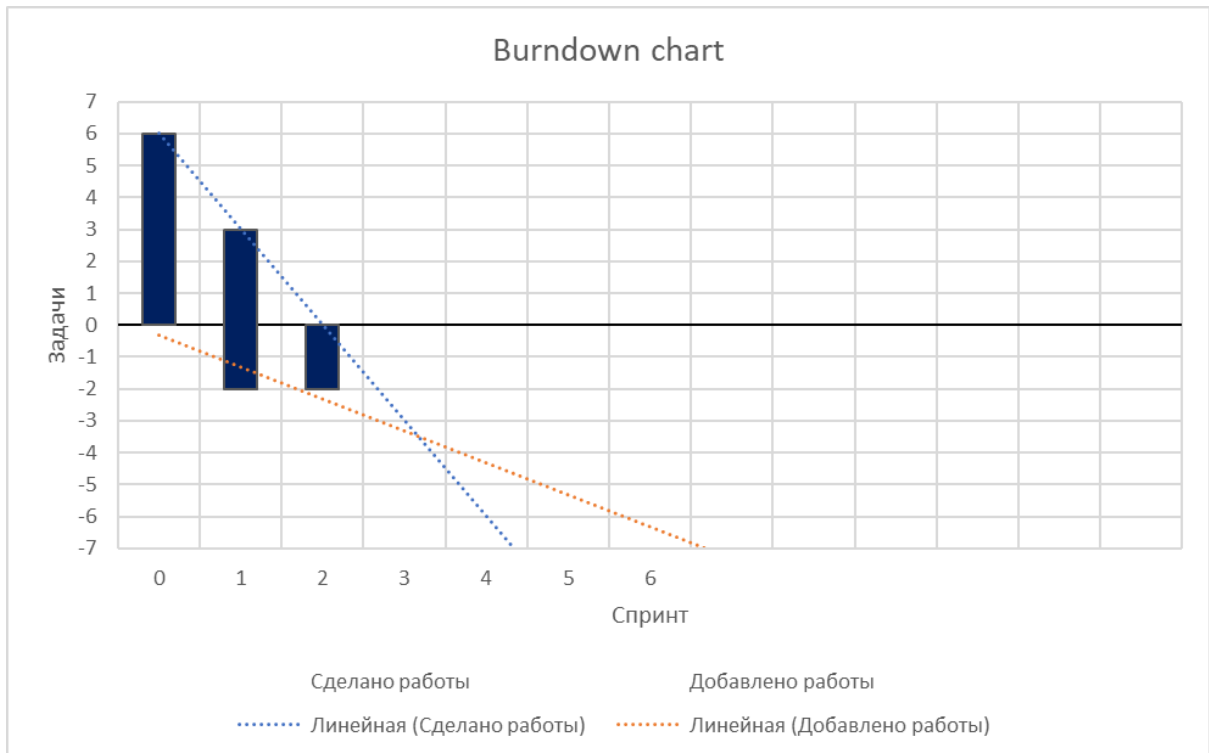
Бэклог, исходные данные, добавим еще выполнение двух задач:

	Сумма всех работ: 8			
№ п/п	Задача	Оценка	В какой спринт пришла	В какой спринт сделали
1	Скачать Docker	1	0	1
2	Выяснить что с виртуализацией, включить ее	2	0	1
3	Установить Docker	1	0	2
4	Скачать и развернуть первый контейнер	2	0	
5	Найти ошибку в запуске Docker, устранить ее	2	1	2

Вычисления отдельные поля и таблица для построения диаграммы:

Исходное количество работы:				6		
Последний спринт:				2		
Средняя скорость выполнения:				3		
Средняя скорость поступления:				1		
Осталось работы:				2		
N	Sprint	Out(n)	In(n)	Сделано работы	Добавле но работы	даты
-2	0	0	0	6	0	
-1	1	3	2	3	-2	
0	2	3	0	0	-2	
1	3			#Н/Д	#Н/Д	
2	4			#Н/Д	#Н/Д	
3	5			#Н/Д	#Н/Д	
4	6			#Н/Д	#Н/Д	
5	7			#Н/Д	#Н/Д	
6	8			#Н/Д	#Н/Д	
7	9			#Н/Д	#Н/Д	

График выгорания задач:



2.6 Рекомендуемая литература

1. Демарко Том, Deadline.
2. Дорофеев Максим, Джедайские техники.