

Отчётность о результатах тестирования

Основы ручного тестирования

Что будет на уроке

1. Метрики покрытия кода.
2. Метрики покрытия требований.
3. Критерии тестирования.
4. Составление отчёта о результатах тестирования в Excel.

Метрики тестирования



Что такое метрики?

Метрика — мера, позволяющая получить численное значение некоторого свойства ПО или его спецификаций.

1. Прогресс относительно запланированного графика и бюджета.
2. Текущее качество объекта тестирования.
3. Адекватность подхода к тестированию.
4. Эффективность активностей тестирования по достижению целей тестирования.

Виды метрик

**Прямые — количественные
показатели**

**Расчётные — рассчитываются
по формуле**

Прямые метрики

Общее количество найденных дефектов — количество дефектов, найденных за весь период тестирования.

Текущее количество дефектов — количество дефектов, найденных в текущем билде, итерации либо спринте.

Плотность дефектов — количество дефектов на единицу размера: на весь продукт, спринт, функцию, объём исходного кода.

Общее устранение дефектов — сколько всего дефектов исправлено за время разработки и тестирования.

Расчётные метрики

Текущее устранение дефектов — процентный показатель устранения в текущем билде либо итерации дефектов, обнаруженных в предыдущем билде или итерации.

Выполнение тест-кейсов — количество выполненных тест-кейсов / общее количество тест-кейсов * 100%.

Стоп-фактор — решение о приостановке тестирования. Учитывается текущее значение метрик, выполнение и успешное прохождение тест-кейсов.

Расчётные метрики

Успешное прохождение тест-кейсов — количество выполненных тест-кейсов / общее количество тест-кейсов * 100%.

Коэффициент ретеста дефектов

- количество закрытых дефектов / количество новых дефектов;
- затраченное время на ретест / затраченное время на тестирование новой функциональности.

Коэффициент регрессии

- количество дефектов в старой инструментарии / количество дефектов в новой инструментарии;
- количество тест-кейсов в статусе failed / количество тест-кейсов в регрессионном наборе.

Метрики тестового покрытия

Покрытие — процентное выражение степени, в которой исследуемый элемент затронут соответствующим набором тест-кейсов.

Тестовое покрытие — плотность покрытия тестами требований или программного кода.

Метрики покрытия кода модульными тест-кейсами — определение, какой процент кода покрыт тестами.

Метрика покрытия требований — требование считается «покрытым», если на него ссылается хотя бы один тест-кейс.

Метрика плотности покрытия требований — учитывается, сколько тест-кейсов ссылается на несколько требований.

Матрица трассировки

Матрица трассировки — это способ для визуализации связей между элементами системы.

Матрица соответствия требований — двумерная таблица, содержащая соответствия функциональных требований продукта и тест-кейсов.

Матрица трассировки

	Тесты	ТК-1	ТК-2	ТК-3
Требования				
Пользователь регистрируется по номеру телефона		+		
Пользователь регистрируется по email			+	
Авторизованный пользователь добавляет товар в корзину				+
Авторизованный пользователь оплачивает заказ картой				
Авторизованный пользователь отменяет заказ				

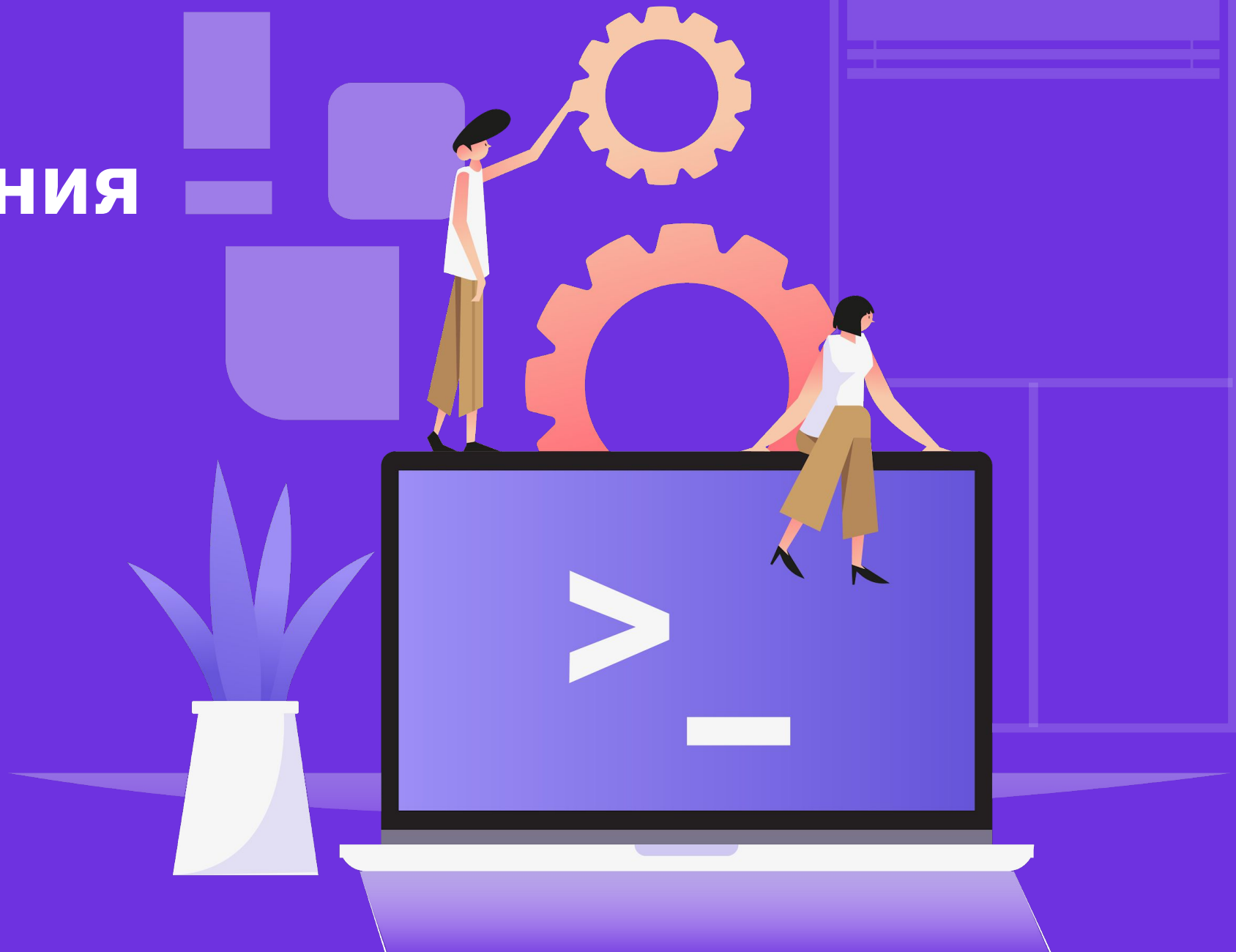
Матрица трассировки

1. Визуализирует актуальное состояние реализации приложения.
2. Декомпозирует и структурирует требования.
3. Показывает, есть ли требования, на которые ещё не запланирована разработка (пропуск реализации).
4. Показывает, реализовано ли требование в конкретный момент.
5. Показывает, покрыто ли требование тест-кейсом (пропуск тестирования).

Этапы составления матрицы трассировки

1. Декомпозиция и приоритизация требований. В результате получается структурированный список атомарных требований.
2. Составление чек-листов и тест-кейсов на основании списка декомпозированных требований.
3. Сопоставление списка требований со списком уже написанных тест-кейсов.
4. Доработка тест-кейсов.
5. Поддержание матрицы трассировки в актуальном состоянии, отслеживание изменений в требованиях и добавление их в таблицу.

Критерии тестирования



Критерии начала тестирования

1. Готовность тестовой платформы.
2. Законченность разработки требуемого инструментария.
3. Наличие требуемой документации.
4. Выход новой версии продукта.
5. Успешный прогон Unit-тестов.
6. Успешный прогон Smoke-тестов.

Критерии приостановки тестирования

1. Критичный баг препятствует тестированию.
2. Накопилось много багов: очевидно, после починки надо всё перепроверять.

Критерии возобновления тестирования

1. Исправление критичного дефекта.
2. Выпуск новой, более стабильной, сборки.

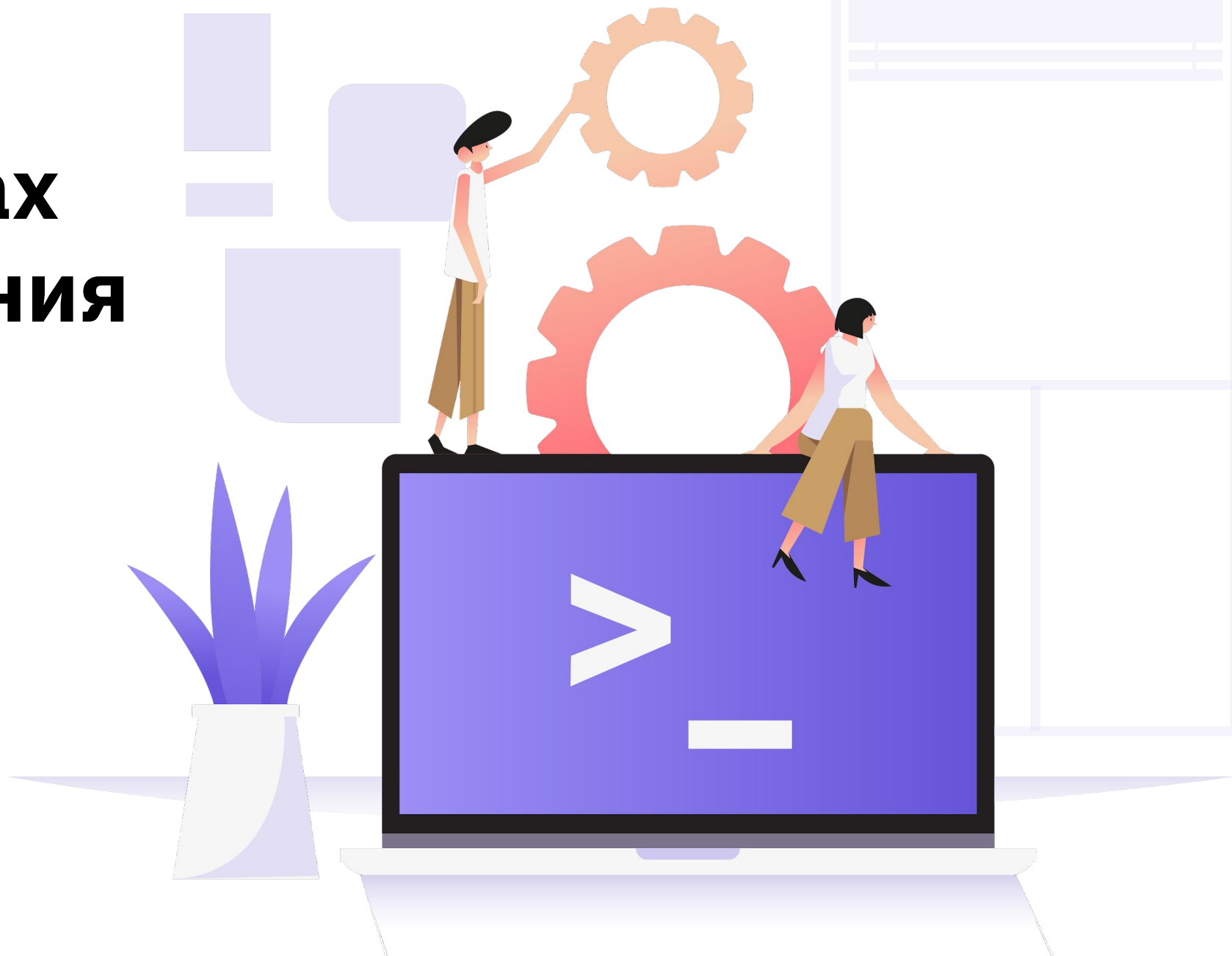
Критерии завершения тестирования

1. Выполнение более 90% запланированных на итерацию тест-кейсов.
2. Выдержка определённого периода без открытия новых дефектов.
3. Истечение времени, отведённого на тестирование.
4. Отсутствие дефектов высокой степени важности.

Приёмочные критерии

1. Успешное прохождение 100% тест-кейсов уровня дымового тестирования (smoke-тестов) и 90% тест-кейсов уровня критического пути.
2. Условия по устранению 100% дефектов критической и высокой важности.
3. Условие, что итоговое покрытие требований тест-кейсами составит не менее 80%.

Отчёт о результатах тестирования



Группы целевых аудиторий

Группа	Приоритеты
Технические пользователи (тест-менеджеры)	<ol style="list-style-type: none">1. Как идёт тестирование?2. Какие проблемы, как решаются?3. Какие используются методы и технологии?
Менеджеры продукта (product owners)	<ol style="list-style-type: none">1. Сроки соблюдаются?2. Какие общие результаты тестирования?3. Какова статистика в целом?
Бизнес-пользователи	<ol style="list-style-type: none">1. Продукт работает?2. Стоит выпускать в промышленную эксплуатацию?3. Какие остались риски?

Глубина временной выборки

Тип отчёта	Особенности
Промежуточный: за день, неделю, спринт и т. д.	<ol style="list-style-type: none">1. Оценка степени готовности продукта.2. Прогресс по сравнению с предыдущим отчётом.
Итоговый: за квартал, по итогам разработки фичи и т. д.	<ol style="list-style-type: none">1. Общий взгляд на проделанную работу.2. Статус продукта в конкретный момент:<ul style="list-style-type: none">• количество неисправленных ошибок;• требуется ли дополнительный цикл тестирования;• оценка возможности выпуска продукта во «внешний мир».

Отчёт о результатах тестирования

1. Краткое описание.
2. Команда тестировщиков.
3. Описание процесса тестирования.
4. Расписание.
5. Статус активностей тестирования и прогресс по сравнению с планом.
6. Факторы, препятствующие прогрессу.
7. Статистика по новым дефектам.
8. Список новых дефектов.
9. Статистика по всем дефектам.
10. Качество объекта тестирования.
11. Рекомендации.
12. Приложения.

Меры повышения качества тестирования

1. Приоритизация тест-комплектов и тест-кейсов.
2. Оптимизация тест-комплектов.
3. Автоматизация регрессионного тестирования.
4. Обучение тестировщиков.
5. Набор новых тестировщиков.

Пример отчёта о тестировании

- [Отчёт по результатам тестирования сайта](#)

Definition of Done

Как ответить на вопрос: «Вы это сделали?»

Интуитивно — есть риск ошибиться, у всех разные представления о том, что значит «сделано».

В соответствии с критериями готовности (definition of done):

- код написан;
- юнит-тесты написаны и успешно выполнены;
- код прошёл ревью;
- документация обновилась;
- функциональное тестирование успешно завершилось;
- регрессионное тестирование успешно завершилось.

Примеры отчётов



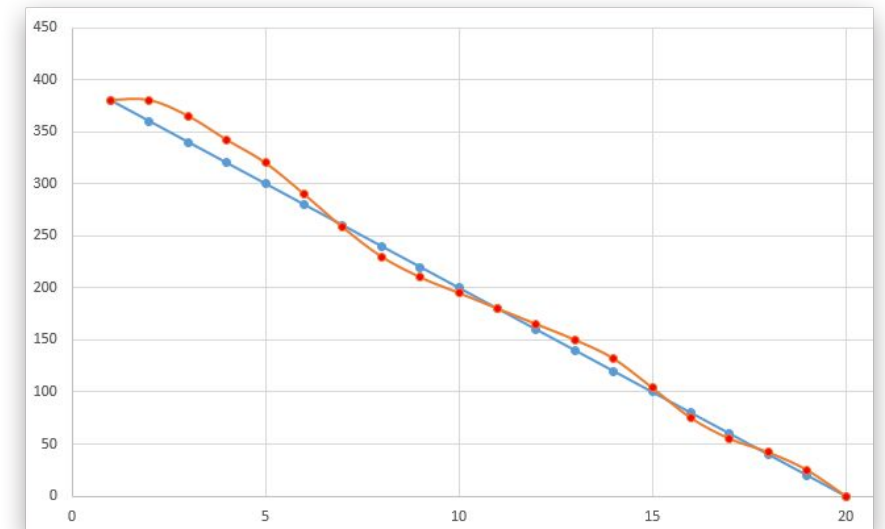
Диаграммы сгорания задач (burndown chart)

Синим цветом — идеальная линия выполнения задач, на которую и следует опираться.

Красным цветом — реальная история выполнения задач.

По шкале Y — количество запланированных баллов (в конкретном случае), идеальные часы, количество задач и так далее.

По шкале X — количество дней до окончания Sprint.

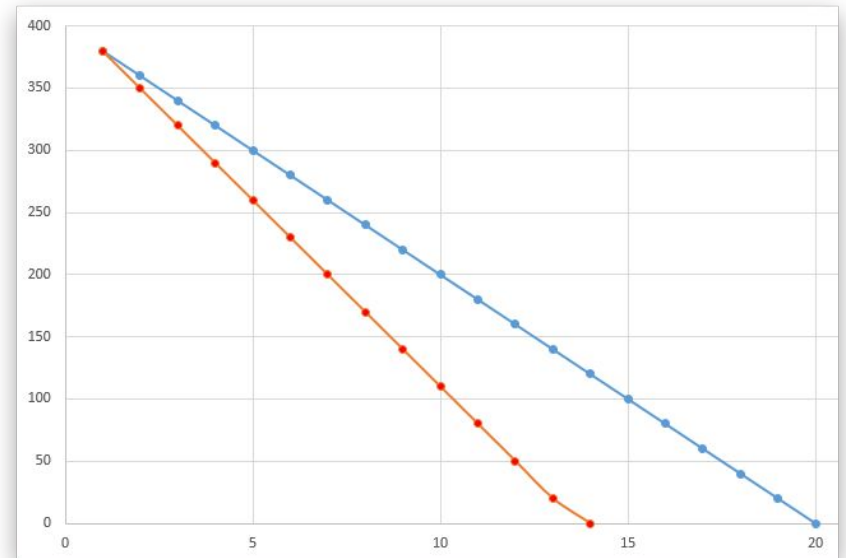


Burndown Chart: Слишком рано

Команда выполнила все задачи раньше срока.

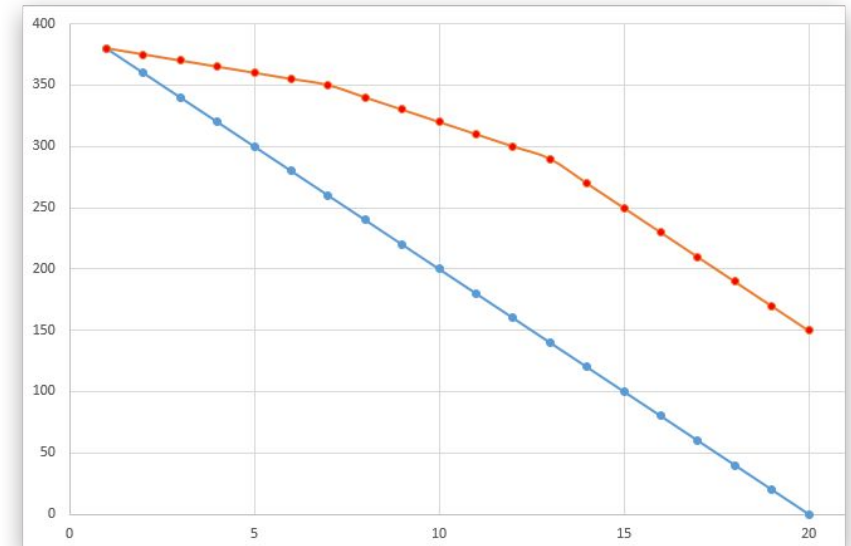
Проблемы:

1. Команда сделала неправильную оценку предстоящей работы.
2. В случае быстрого выполнения задач разработчики не добавляли задачи из следующего спринта.
3. Команда сильно перестраховалась, включив изначально дополнительный срок.

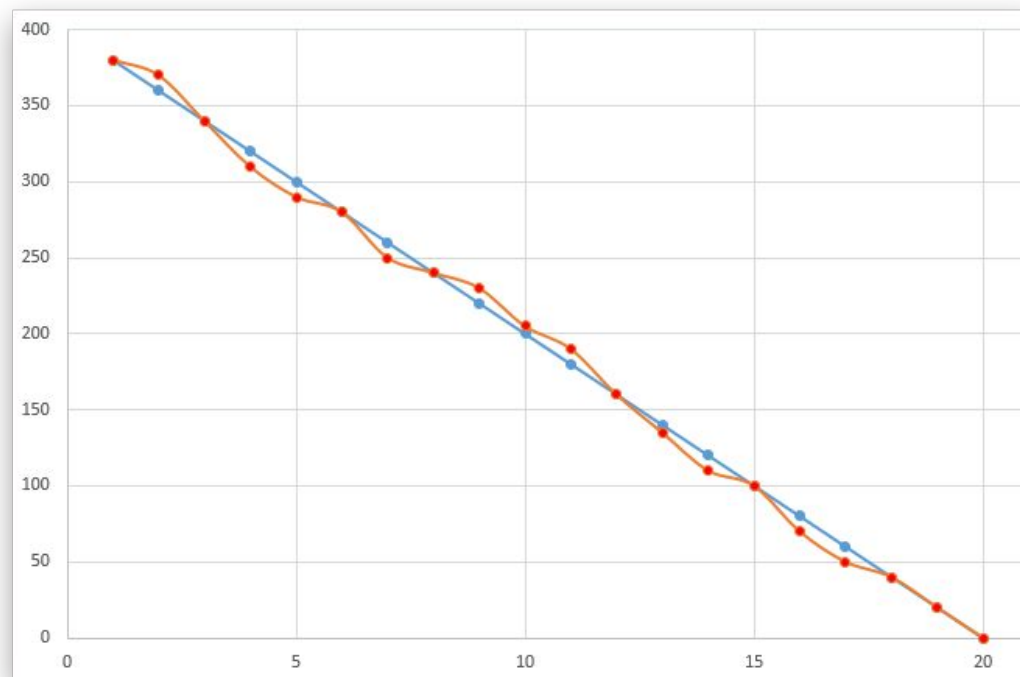


Burndown Chart: Опоздали

1. Постоянное добавление новых задач во время спринта увеличило нагрузку.
2. Недоделанность задач, когда задачи сделаны наполовину.



Burndown Chart: Идеально



Как построить диаграмму Burndown

1. Определить общее количество задач.
2. Определить количество дней на выполнение всех задач.
3. Вычислить среднее количество задач на 1 день (план).
4. Заполнить таблицу плановыми и фактическими значениями.
5. Построить график.

Создание отчёта в Excel

[Файл с данными](#)

Задачи:

1. Построить диаграмму burndown.
2. Построить диаграмму по прохождению тестов:
 - общую;
 - в разрезе исполнителей.
3. Общий статус по всем тест-кейсам — сколько тестов и в каком статусе.
4. Динамика по багам:
 - сколько открыли (по дням) — график;
 - сколько починили (по дням) — график;
 - сколько всего багов в разрезе статусов — диаграмма;
 - сколько всего багов в разрезе критичности — диаграмма.

Спасибо!
Каждый день
вы становитесь
лучше :)

