ОГЛАВЛЕНИЕ

1. Качество тестирования.....
1.1 Пропуски дефектов .....
1.2 Тестовое покрытие – 1/2 .....
1.2 Тестовое покрытие – 2/2 .....
1.3 Качество тест-дизайна.....
2. Проектное планирование .....
2.1 Следование плану работ .....
2.2 Учёт проектных рисков.....
2.3 Метрики для прогнозирования трудозатрат по тестированию .....
3. Качество продукта.....
3.1 Удовлетворение пользователей .....
3.2 Дефекты в продукте .....
3.3 Результаты тестирования .....
3.4 Характеристики качества ПО – 1/4 .....
3.4 Характеристики качества ПО – 2/4 .....
3.4 Характеристики качества ПО – 3/4 .....
3.4 Характеристики качества ПО – 4/4 .....
4. Эффективность тестирования .....
4.1 Скорость тестирования.....
4.2 Финансовые показатели.....
4.3 Работа с дефектами.....
4.4 Автотесты.....

Метрики. Самое важное

Никакие метрики не являются универсальными!
Это лишь инструмент, который помогает вам в решении определенных задач.
Сначала мы определяем цели, и только потом думаем, какие метрики могут нам помочь в их достижении (и могут ли).

Для чего вам нужны метрики?

1. **Оценка прогресса**. Если перед нами стоит какая-то задача, которую невозможно выполнить “здесь и сейчас”, нам нужен инструмент оценки, чтобы понять, ведут ли наши действия к ожидаемому результату?

2. **Промежуточные замеры.** Есть метрики, показывающие финальный результат, а есть процессные, благодаря которым еще до релиза продукта и до выдачи отчета руководству мы можем определить, движемся ли мы в нужном направлении и что еще нам нужно улучшать.

3. **Поиск проблем.** Третий случай, когда мы используем метрики, – это проведение аудита и поиск слабых мест. Тут бывают два стандартных сценария: либо мы ищем проблему, которую пока не можем осознать, либо ищем корни и первопричины у известных проблем.

4.**Числовые обоснования.** Ну, и последняя цель, которую мы можем преследовать при внедрении метрик, – это презентация и иллюстрация руководству. Через метрики мы можем конструктивно и наглядно обосновать то, что почти невозможно донести на уровне простого общения: потребность в ресурсах, проблемы в разработке, влияние недостающих требований на общий процесс разработки и т.д.

Метрики: как выбрать и внедрить

Всегда идите в направлении **от задачи к метрике**, а не наоборот. Например:

- Согласуйте ожидания руководства, зафиксируйте, а потом уже думайте: какими показателями можно оценить именно эту цель?
- Вы хотите что-то улучшить в своей работе. Что именно? Определившись, какой результат вы хотите достигнуть, подумайте о его измерении:
  – как оценивать прогресс в достижении этого показателя?
  – как измерять достижения на уровне процесса, пока ключевой показатель, возможно, не меняется?
- Вы хотите решить какую-то проблему, непосредственно в команде тестирования или на проекте в целом:
  – как оценивать эту проблему, в чём её можно измерить?
  – какие могут быть предпосылки для этой проблемы, очевидные или кажущиеся совсем нелогичными?
  – как можно обнаружить корень этой проблемы, или “узкое горлышко”?
- Вам надо обосновать что-то своему руководству, и вы исчерпали аргументы:
  – как проиллюстрировать наличие проблемы?
  – как показать пользу от предлагаемого вами решения?

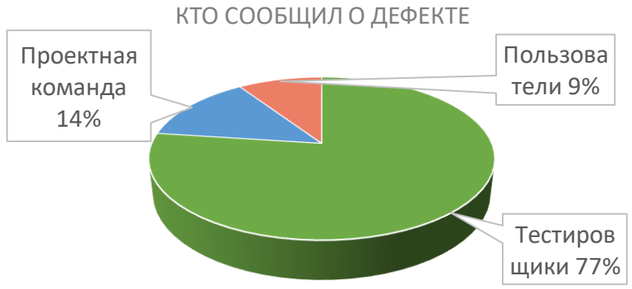
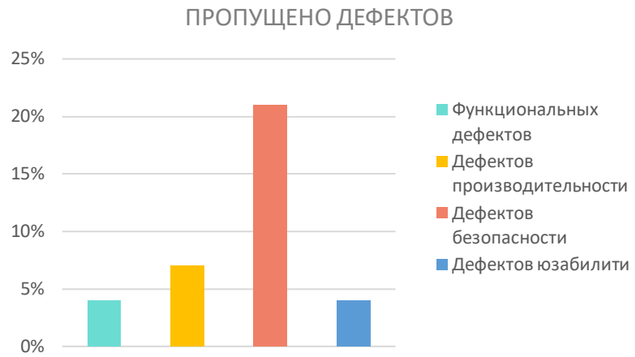
Максимально осознанно пройдитесь по этому списку вопросов. Подключите коллег через совместный брейншторм или опросы. Определите, что вы хотите измерять, не доходя до уровня конкретных чисел и метрик.

Сначала проведите работу по анализу ваших потребностей в измерении, и только после этого ознакомьтесь с примерами метрик.

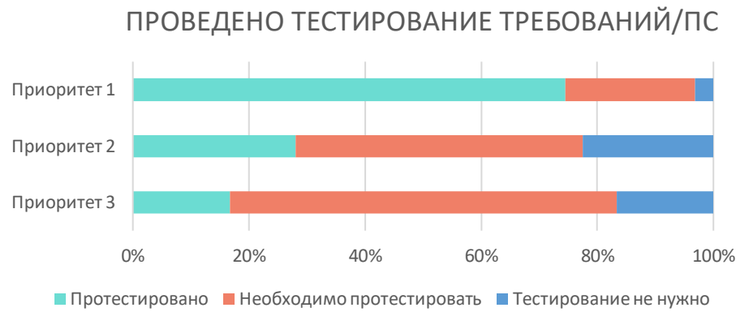
Скорее всего в них вы найдёте такие показатели, которые помогут в решении стоящих перед вами задач. Но что делать, если необходимость в метрике вы выявили, но подходящего варианта для её расчёта не нашли? Пишите нам! Обещаем по каждому запросу на вариант внедрения метрики подготовить такой способ расчёта и визуализации, который получится внедрить в ваши условия. Таким образом, постепенно мы сделаем этот документ ещё более полным и полезным для всей отрасли.

1. Качество тестирования

1.1 Пропуски дефектов

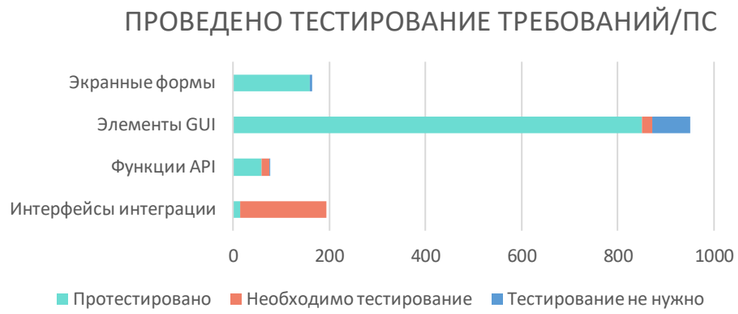
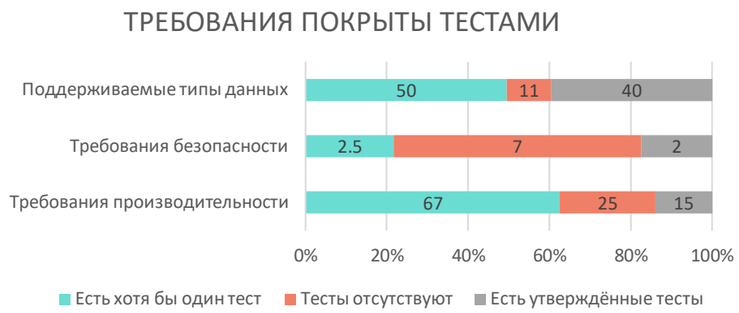
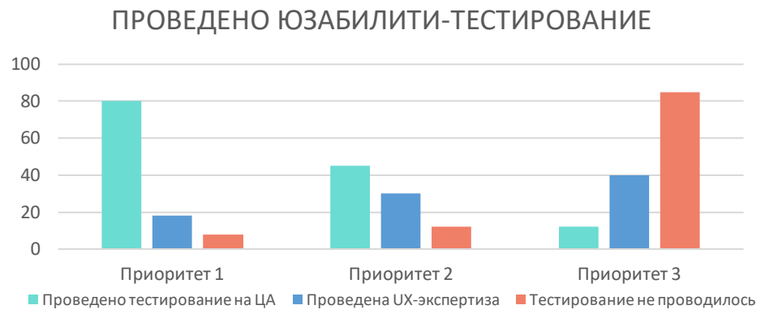
#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
1	Количество дефектов, пропущенных в продуктив	Bugs Leakage	$\frac{\text{Дефектов, зарегистрированных на пром. среде}}{\text{Всего зарегистрировано дефектов}} \times 100\%$ Для сбора метрики необходимо добавить поле в баг-трекер «на каком окружении был обнаружен дефект»		• Как один из KPI тестирования • Для анализа причин ненахождения дефектов в тестовой сборке
2	% дефектов, найденных пользователями	Bugs Reported by Users	$\frac{\text{Дефектов выявлено пользователями}}{\text{Всего зарегистрировано дефектов}} \times 100\%$		• Как один из KPI тестирования • Для анализа, какие дефекты критичны для пользователей
3	% дефектов, найденных не тестировщиками	Bugs Reported not by Testers	$\frac{\text{Дефектов, выявленных не тестировщиками}}{\text{Всего зарегистрировано дефектов}} \times 100\%$		• Как один из KPI тестирования • Для анализа, почему другие участники процесса находят ошибки, пропущенные тестировщиками
4	Пропуски дефектов по категориям	Bugs Leakage by Category	$\frac{\text{Дефектов с пром среды по категории}}{\text{Всего дефектов по категории}} \times 100\%$		Для анализа, в каких областях необходимо развивать тестирование • В качестве категорий могут выступать уровни тестирования, типы тестирования, зоны функциональности и т.д.

1.2 Тестовое покрытие – 1/2

<table>
<tr><th>#</th><th>Метрика рус.</th><th>Метрика англ.</th><th>Как рассчитывать</th><th>Визуализация метрики</th><th>Как использовать</th></tr>
<tr><td>5</td><td>Покрытие требований тестами</td><td>Requirements Coverage</td><td>$$\frac{\text{Требований с тестами}}{\text{Всего требований}} \times 100\%$$
Для расчёта этой метрики необходимо определить критерий «требований с тестами». Это может быть «хотя бы 1 тест», «хотя бы 1 тест на каждую границу и т.д.». Рассчитывается в системе ведения требований, по статусу или по наличию ссылки на тесты.</td><td rowspan="2">
</td><td rowspan="2">• Для планирования расширения тестового покрытия
• Для оценки рисков пропуска дефектов</td></tr>
<tr><td>6</td><td>Подтверждённое покрытие требований тестами</td><td>Approved Requirements Coverage</td><td>$$\frac{\text{Требований с утверждёнными тестами}}{\text{Всего требований}} \times 100\%$$
Для расчёта этой метрики необходимо согласование тестовых наборов по каждому требованию. Чаще всего такое согласование происходит с аналитиком и разработчиком, ответственным за реализацию требования. По итогам обсуждения в системе ведения требований проставляется статус «тесты согласованы»</td></tr>
<tr><td>7</td><td>Покрытие требований исследовательскими тестами</td><td>ET Requirements Coverage</td><td rowspan="2">Постановка задач на исследовательское тестирование по требованиям или пользовательским сценариям и расчёт % требований, по которым было проведено тестирование.</td><td rowspan="2"> 
</td><td rowspan="2">• Для оценки готовности к релизу
• Для планирования затрат на тестирование</td></tr>
<tr><td>8</td><td>Покрытие пользовательских сценариев тестами</td><td>User Stories Test Coverage</td></tr>
<tr><td>9</td><td>Покрытие кода по функциям</td><td>Code Coverage</td><td rowspan="4">• Выбор инструмента оценки, исходя из потребностей и архитектуры проекта
• Создание инструментальной сборки
• Проведение тестов на этой сборке для анализа покрытия

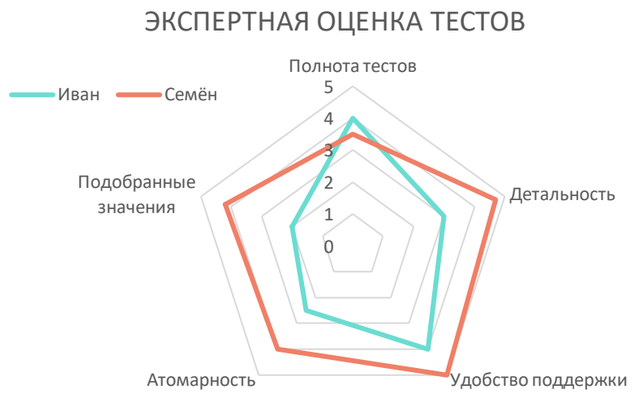
Оценка покрытия кода возможна как для автоматизированных, так и для ручных тестов!</td><td rowspan="4">All files
74,83% Statements 28,33% Branches 77,55% Functions 76,12% Lines
<table>
<tr><th>File</th><th>Statements</th><th></th><th>Branches</th><th></th><th>Functions</th><th></th><th>Lines</th><th></th></tr>
<tr><td>src</td><td>100%</td><td>3/3</td><td>100%</td><td>0/0</td><td>100%</td><td>0/0</td><td>100%</td><td>3/3</td></tr>
<tr><td>src/app</td><td>100%</td><td>4/4</td><td>100%</td><td>0/0</td><td>100%</td><td>0/0</td><td>100%</td><td>3/3</td></tr>
<tr><td>src/app/edit</td><td>67,74%</td><td>21/31</td><td>25%</td><td>1/4</td><td>66,67%</td><td>6/9</td><td>65,52%</td><td>19/29</td></tr>
<tr><td>src/app/search</td><td>94,74%</td><td>18/19</td><td>50%</td><td>2/4</td><td>87,50%</td><td>7/8</td><td>93,33%</td><td>14/15</td></tr>
<tr><td>src/app/shared/search</td><td>67,65%</td><td>23/34</td><td>15,63%</td><td>5/32</td><td>80%</td><td>12/15</td><td>65,63%</td><td>21/32</td></tr>
</table></td><td rowspan="4">• Для оценки уровня покрытия
• Для исследования влияющих параметров, не обозначенных в документации, но влияющих на выполнение кода продукта
• Для расширения тестового покрытия</td></tr>
<tr><td>10</td><td>Покрытие кода по условиям</td><td>Alternatives Coverage</td></tr>
<tr><td>11</td><td>Покрытие кода по решениям</td><td>Decision Coverage</td></tr>
<tr><td>12</td><td>Покрытие строк кода</td><td>Path Coverage</td></tr>
</table>

1.2 Тестовое покрытие – 2/2

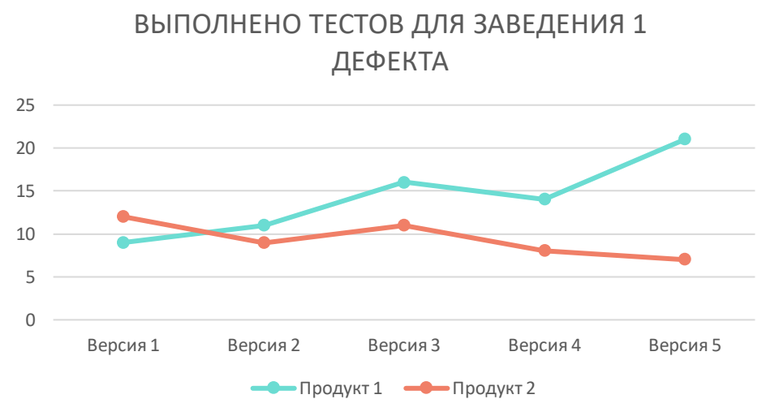
<table>
<tr><th>#</th><th>Метрика рус.</th><th>Метрика англ.</th><th>Как рассчитывать</th><th>Визуализация метрики</th><th>Как использовать</th></tr>
<tr><td>13</td><td>Покрытие GUI</td><td>GUI Coverage</td><td rowspan="3">$\frac{\text{Объектов покрыто тестами}}{\text{Всего объектов}} \times 100\%$
Где в качестве объектов могут выступать:
• Экранные формы
• Элементы графического интерфейса
• Функции API
• Интерфейсы интеграции
• И т.д.</td><td rowspan="3"> 
</td><td rowspan="3">• Выявить зоны рисков и «узкие горлышки»
• Оценить статус тестирования</td></tr>
<tr><td>14</td><td>Покрытие API</td><td>API Coverage</td></tr>
<tr><td>15</td><td>Покрытие интеграций</td><td>Integration Coverage</td></tr>
<tr><td>16</td><td>Покрытие требований производительности</td><td>Performance Coverage</td><td rowspan="3">$\frac{\text{Требований по группе покрыто тестами}}{\text{Всего требований в группе}} \times 100\%$
Где в качестве группы могут выступать:
• Разные типы требования
• Разные модули
• Разные функциональные области
• И т.д.</td><td rowspan="3"> 
</td><td rowspan="3">• Для планирования расширения тестового покрытия
• Для оценки рисков пропуска дефектов</td></tr>
<tr><td>17</td><td>Покрытие требований нагрузки</td><td>Load Coverage</td></tr>
<tr><td>18</td><td>Покрытие поддерживаемых форматов данных</td><td>Data Coverage</td></tr>
<tr><td>19</td><td>Покрытие пользовательских сценариев юзабилити-тестами</td><td>Use-Cases Covered by Usability Tests</td><td>• Проведение юзабилити-экспертизы с привлечением ЦА с анализом возможности выполнения сценариев
• Проведение внутренней UX-экспертизы по удобству выполнения пользовательских сценариев</td><td> 
</td><td>• Оценить юзабилити-риски
• Принять решение, какие тесты необходимо провести</td></tr>
<tr><td>20</td><td>Покрытие окружений уникальными тестами</td><td>Environmental Unique Tests Coverage</td><td>• Анализ рисков, связанных с окружениями
• Подготовка тестов, зависящих от окружения
• Согласование достаточности тестов, зависящих от окружения</td><td><table>
<tr><th>Платформа</th><th>Windows</th><th>Mac</th><th>Android</th><th>Unix</th></tr>
<tr><td>Применимо общих тестов</td><td>1350</td><td>1350</td><td>340</td><td>940</td></tr>
<tr><td>Возможно выполнить только на этой платформе</td><td>42</td><td>60</td><td>160</td><td>36</td></tr>
<tr><td>Уникальные риски на этой платформе</td><td>28</td><td>n/a</td><td>112</td><td>n/a</td></tr>
<tr><td>Согласованы уникальные тесты с командой разработки</td><td>X</td><td>X</td><td>V</td><td>X</td></tr>
</table></td><td>• Оценить качество планирования тестов, зависящих от окружения
• Определить необходимость дополнительного анализа
• Запланировать расширение тестового покрытия</td></tr>
<tr><td>21</td><td>Проведение тестов на поддерживаемых окружениях</td><td>Environments Coverage</td><td>$\frac{\text{Проведено тестов на платформе}}{\text{Всего тестов отобрано для платформы}} \times 100\%$</td><td><table>
<tr><th>Платформа</th><th>Windows</th><th>Mac</th><th>Android</th><th>Unix</th></tr>
<tr><td>Всего тестов</td><td>1420</td><td>1438</td><td>612</td><td>1003</td></tr>
<tr><td>Проведено тестов 1 прио</td><td>92%</td><td>27%</td><td>96%</td><td>14%</td></tr>
<tr><td>Проведено тестов 2 прио</td><td>86%</td><td>13%</td><td>100%</td><td>6%</td></tr>
</table></td><td>• Оценить риски, связанные с окружениями
• Запланировать дополнительные тесты окружения</td></tr>
</table>

1.3 Качество тест-дизайна

<table>
<tr><th>#</th><th>Метрика рус.</th><th>Метрика англ.</th><th>Как рассчитывать</th><th>Визуализация метрики</th><th>Как использовать</th></tr>
<tr><td>22</td><td>Средняя экспертная оценка тестов</td><td>Expert test cases evaluation</td><td>AVG
(Оценки тестов в системе хранения тестов)

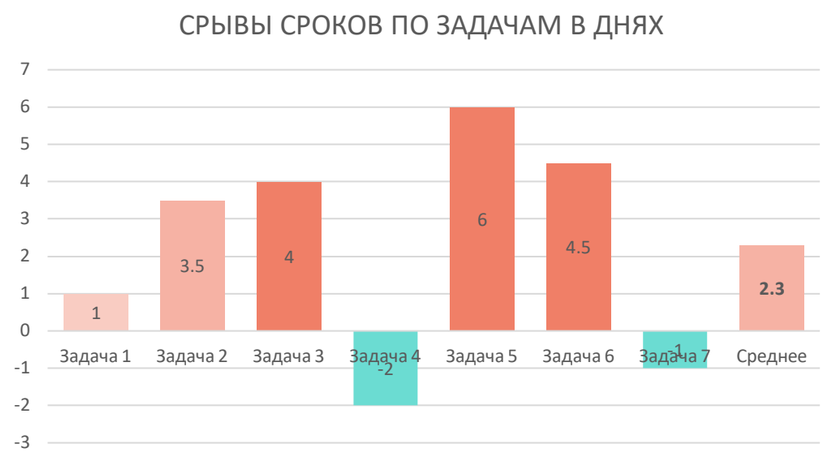
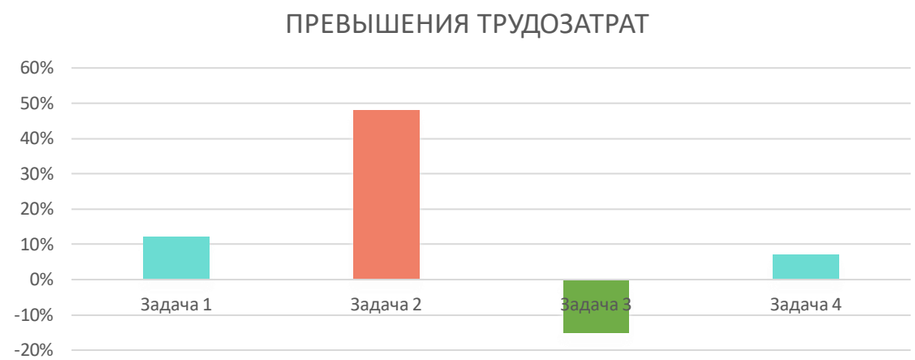
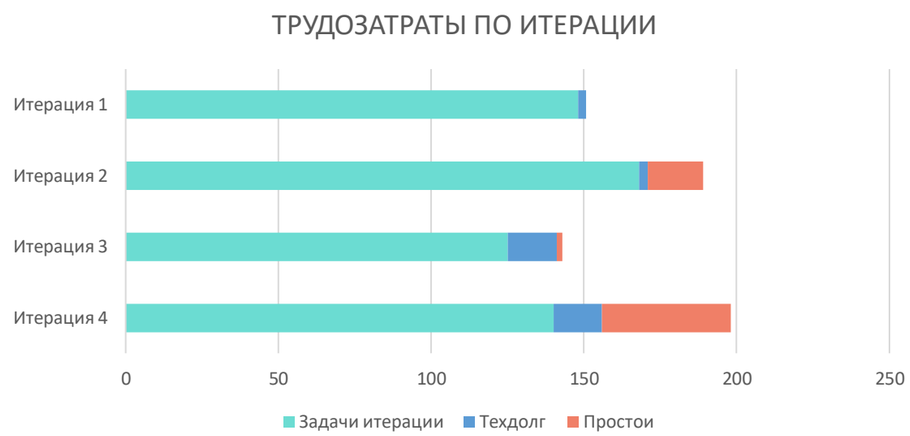
Тесты можно оценивать:
Одной оценкой
По различным шкалам
Оценки можно категоризировать:
По сотрудникам
По группам
По типам или областям тестов
И т.д.</td><td> 
</td><td>Для выявления слабых зон в тестовом покрытии
Для обнаружения нехватки квалификации и принятия решений о дополнительном обучении
Для выявления тестов, которые необходимо улучшить</td></tr>
<tr><td>23</td><td>Актуальность тестов</td><td>Test cases relevance</td><td>AVG (Последняя версия документации - Последняя версия документации, использованная в тестах)
или
$\frac{\text{Тестов, акт. в последней версии}}{\text{Общее число тестов}} \times 100\%$</td><td><table><tr><th>Модуль</th><th>% актуальных тестов</th><th>Среднее отклонение версии от последней</th></tr><tr><td>Модуль 1</td><td>100%</td><td>0</td></tr><tr><td>Модуль 2</td><td>46%</td><td>2,6</td></tr><tr><td>Модуль 3</td><td>89%</td><td>0,33</td></tr></table></td><td>Оценить риски использования текущих тестов
Определить необходимость выделения ресурсов на актуализацию тестов</td></tr>
<tr><td>24</td><td>Рейтинг обнаружения дефектов тестами</td><td>TC bugs detecting ratio</td><td>$\frac{\text{Выполнено тестов}}{\text{Зарегистрировано дефектов}}$

Считается по версии, итерации или периоду времени

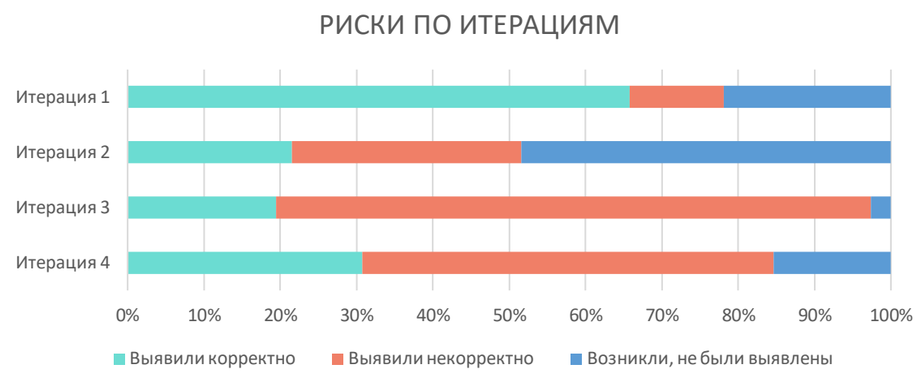
Возможна категоризация статистики по:
По сотрудникам (тест-дизайнерам)
По зонам функциональности
По типам тестов
И т.д.</td><td> 
</td><td>Выявить тесты, подверженные эффекту пестицида (см. Продукт 1)
Оценить эффективность внедрения новых техник и подходов (см. Продукт 2)</td></tr>
<tr><td>25</td><td>Скорость обнаружения дефектов по тест-кейсам</td><td>TC bugs detection speed</td><td>$\frac{\text{Зарегистрировано дефектов}}{\text{Затрачено времени на тестирование по ТК}}$</td><td><table><tr><th>Проект</th><th>Проект 1</th><th>Проект 2</th></tr><tr><td>Затраты на исследовательское тестирование</td><td>140 ч/ч</td><td>166 ч/ч</td></tr><tr><td>Багов найдено в ИТ</td><td>73</td><td>91</td></tr><tr><td>Затраты на обнаружение 1 бага в ИТ</td><td>1,9 ч/ч</td><td>1,8 ч/ч</td></tr><tr><td>Затраты на проверку тест-кейсов</td><td>97 ч/ч</td><td>130 ч/ч</td></tr><tr><td>Багов найдено по ТК</td><td>132</td><td>48</td></tr><tr><td>Затраты на обнаружение 1 бага по ТК</td><td>0,7 ч/ч</td><td>2,7 ч/ч</td></tr></table></td><td rowspan="2">Выявить эффективные тесты, скорость обнаружения ошибок, по которым выше, чем в ИТ
Выявить неэффективные тесты, по которым находится мало дефектов</td></tr>
<tr><td>26</td><td>Эффективность тест-кейсов по сравнению с исследовательским тестированием</td><td>Exploratory / Scripted Testing Efficacy Comparison</td><td>$\frac{\text{Скорость обнаружения по ТК}}{\text{Скорость обнаружения в ИТ}}$</td><td><table><tr><th>Проект</th><th>Проект 1</th><th>Проект 2</th></tr><tr><td>Соотношение эффективности ИТ и ТК</td><td>2,7</td><td>0,6</td></tr></table></td></tr>
</table>

2. Проектное планирование

2.1 Следование плану работ

#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
27	Срывы сроков по задачам	Schedule slippage	Дата финиша факт – Дата финиша план Метрика может собираться по: • Задачам • Итерациям • Релизам И категоризироваться по: • Сотрудникам • Типам задач • И т.д.		• Для выявления среднего срыва сроков и включения его в планы работ • Для анализа причин сроков по каждой задаче или по самым большим сдвигам • Для включения в планирование рисков исходя из статистики срывов в процентах
28	Отклонение от плана работ	Schedule Variance	При оценке групп задач, итераций и проектов оценивается в процентах: $\frac{\text{Дата фишина факт} - \text{Дата финиша план}}{\text{Дата финиша план} - \text{Дата старта план}} \times 100\%$		
29	Превышение трудозатрат	Estimation Changes	$\frac{\text{Фактические трудозатраты}}{\text{Плановые трудозатраты}} \times 100\% - 100\%$ Данные можно категорировать по: • Задачам • Итерациям • Командам • Сотрудникам • И т.д.		• Для анализа причин отклонения в оценках • Как возможный KPI при необходимости снижения трудозатрат • Для поиска превышения трудозатрат по категориям
30	Простои суммарно	Team Idling	В случае, если специалист не может выполнять задачи по текущей итерации (заблокирована сборка, не готово тестовое окружение и т.д.), то он списывает затраты этого периода в категорию «Простои»		• Для анализа причин простоев • Для закладывания простоев в планы работ • Для контроля за динамикой при борьбе с простоями
31	Доля простоев	Team Idling %	$\frac{SUM\ (\text{списания по простоям за период})}{SUM\ (\text{все списания за период})} \times 100\%$		

2.2 Учёт проектных рисков

<table>
<tr><th>#</th><th>Метрика рус.</th><th>Метрика англ.</th><th>Как рассчитывать</th><th>Визуализация метрики</th><th>Как использовать</th></tr>
<tr><td>32</td><td>Ведение рисков на проекте</td><td>Risk management</td><td>Наличие процесса ведения рисков на проекте</td><td rowspan="2">
<table>
<tr><th>Проект</th><th>Учёт рисков</th><th>Стратегия нивелирования</th><th>Согласование стратегии</th></tr>
<tr><td>Проект 1</td><td>V</td><td>X</td><td>X</td></tr>
<tr><td>Проект 2</td><td>V</td><td>V</td><td>V</td></tr>
<tr><td>Проект 3</td><td>V</td><td>V</td><td>X</td></tr>
<tr><td>Проект 4</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>Проект 5</td><td>V</td><td>X</td><td>X</td></tr>
</table>
</td><td rowspan="2">
Для оценки уровня зрелости проектов
Для анализа, где необходимо внедрение учёта рисков для более грамотного планирования
</td></tr>
<tr><td>33</td><td>Командное согласование рисков</td><td>Risk management approval</td><td>Согласование рисков и стратегии нивелирования всей командой проекта</td></tr>
<tr><td>34</td><td>Корректность прогнозирования рисков</td><td>Risk predicting correctness</td><td>$$\frac{\text{Возникшие риски по прогнозу}}{\text{Все риски}} \times 100\%$$
Где «Все риски» это сумма:
• Спрогнозированных возникших
• Спрогнозированных не возникших
• Возникших не спрогнозированных</td><td> 
</td><td>
Выявление избыточного прогнозирования рисков (см. Итерация 3)
Выявление недостаточного прогнозирования (см. Итерация 2)
</td></tr>
<tr><td>35</td><td>Уровень нивелирования рисков</td><td>Risk avoidance efficacy</td><td>$$\frac{\text{Успешно нивелированы}}{\text{Все риски в стратегии}} \times 100\%$$
Рассчитывается на основании стратегии нивелирования</td><td>
<table>
<tr><th>Проект</th><th>Проект 1</th><th>Проект 2</th><th>Проект 3</th><th>Проект 4</th></tr>
<tr><td>Рисков выявлено</td><td>42</td><td>35</td><td>28</td><td>70</td></tr>
<tr><td>Рисков со стратегией решения</td><td>42</td><td>16</td><td>26</td><td>56</td></tr>
<tr><td>Рисков, нивелированных по стратегии</td><td>14</td><td>3</td><td>20</td><td>18</td></tr>
<tr><td>Рисков, стратегия нивелирования по которым не помогли</td><td>6</td><td>11</td><td>3</td><td>7</td></tr>
<tr><td>Уровень нивелирования</td><td>70%</td><td>21%</td><td>87%</td><td>72%</td></tr>
</table>
</td><td>
Оценить, насколько корректные решения мы выбираем для нивелирования рисков
Найти более эффективные инструменты нивелирования
</td></tr>
</table>

2.3 Метрики для прогнозирования трудозатрат по тестированию

#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
36	Соотношение трудозатрат разработчиков и тестировщиков	Test/Dev Effort Rate	$\frac{\text{Затраты на тестирование}}{\text{Затраты на разработку}}$ Рассчитывается по доработке, итерации, модулю и т.д.		
37	Затраты ч/ч на KLOC	Test Effort per KLOC	$\frac{\text{Затраты на тестирование доработки}}{\text{Объём кода доработки}}$ Рассчитывается по объёму кода в строках (KLOC) или с учётом сложности кода (Cyclomatic Complexity)		
38	Затраты ч/ч на проверку требования	Test Effort per Req	$\frac{\text{Затраты на тестирование доработки}}{\text{Число требований в доработке}}$		
39	Затраты на 1 тестовое окружение	Tet Effort per 1 Environment	$\frac{\text{Затраты на тестирование совместимости}}{\text{Протестировано окружений}}$		
40	Время, затрачиваемое на создание ТК/ЧЛ	Test Effort for 1 TC creation	$\frac{\text{Затраты на создание тестов}}{\text{Число созданных тестов}}$		
41	Затраты на проверку 1 инцидента	Test Effort for 1 incident submit	$\frac{\text{Затраты на проверку инцидентов за период}}{\text{Число инцидентов обработано}}$		
42	Затраты на валидацию 1 дефекта	Test Effort for 1 bug validation	$\frac{\text{Затраты на валидацию дефектов}}{\text{Число дефектов провалидировано}}$		

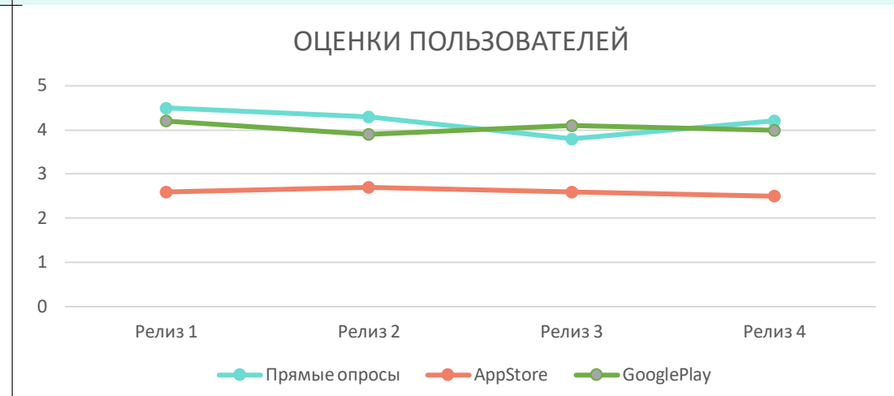
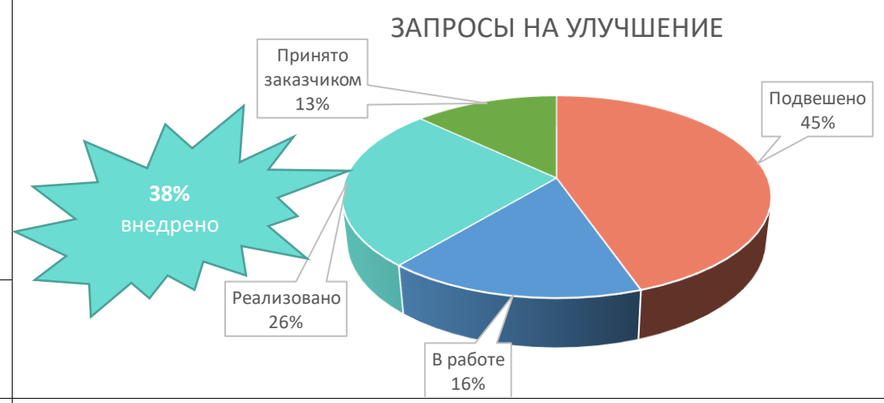
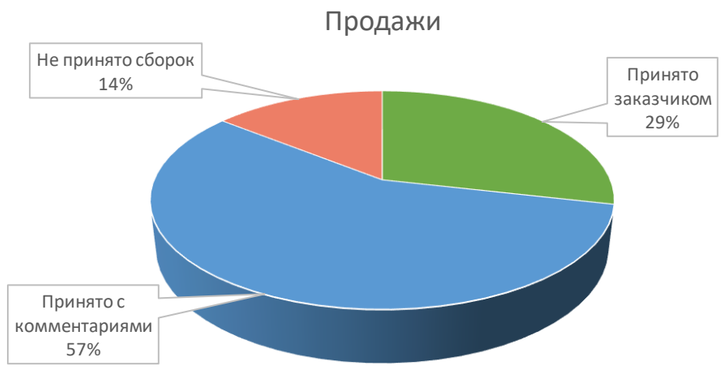
Статистика по доработке	Доработка 1	Доработка 2	Доработка 3	Доработка 4	Среднее
Затраты на тестирование	168	143	224	130	166
Затраты на разработку	353	279	482	273	347
Соотношение затрат	2.1	1.95	2.15	2.1	2.1
CLOK	16	14.5	20	13	15.9
Затрат на CLOK	10.5	9.9	11.2	10	10.4
Число требований	111	98	192	104	126
Затрат на 1 требование	1.5	1.5	1.2	1.25	1.36
Затраты на проведение тестов совместимости на 1 окружении	12	12	11	12	11.5
Затраты на проверку инцидентов	6	15	8	3	8
Число инцидентов	17	32	21	8	19.5
Затраты на 1 инцидент	0.35	0.47	0.38	0.38	0.395
Затраты на разработку тестов	32	28	43	23	31.5
Число тестов	140	125	281	116	165.5
Затраты на 1 тест	0.23	0.22	0.16	0.2	0.2
Затраты на валидацию дефектов	11	9.5	17	8	11.375
Число дефектов провалидировано	46	39	64	36	46.25
Затраты на валидацию 1 дефекта	0.24	0.24	0.27	0.22	0.24

Как использовать:

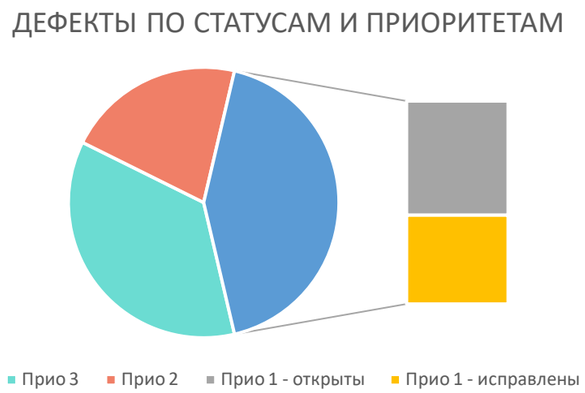
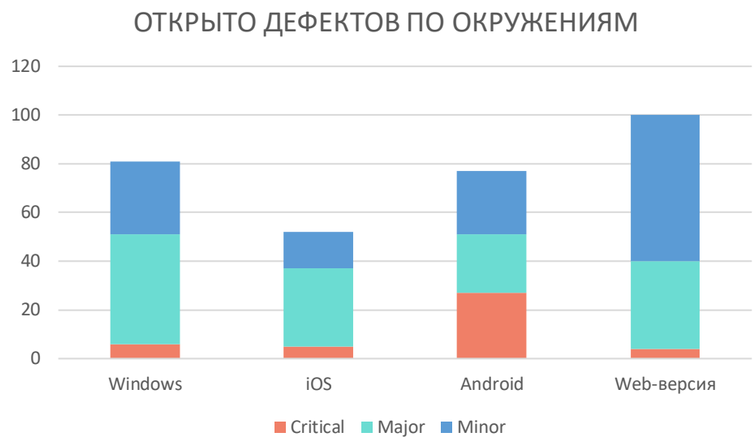
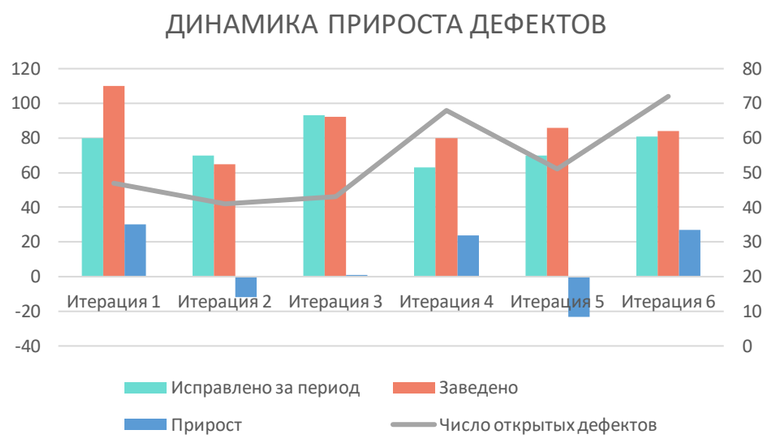
- Метрики грубой оценки (36,37,38) используются для быстрого получения планируемых трудозатрат исходя из предварительно собранной статистики. Если на тестирование поступает доработка, содержащая 150 требований, мы можем быстро оценить затраты на её тестирование как 150x1.36 = 204 ч/ч.
- Метрики точной оценки используются при планировании трудозатрат по стратегии тестирования. В стратегии мы указываем, что нам необходимо создать около 120 тестов, провести тестирование на 4 тестовых окружениях, провалидировать 40 дефектов, и т.д. В оценке мы суммируем затраты на каждый из этих пунктов, затраты берём исходя из предварительно собранной статистики.
- Анализируя затраты на повторяющиеся активности, мы можем выявить, что некоторые типы задач мы выполняем слишком долго. В этом случае мы проводим улучшения и оптимизацию процесса, направленные на сокращение затрат, и выносим такие показатели в KPI (например, цель – тратить на валидацию 1 дефекта не более 10 минут, стратегия – автоматизация создания тестовых данных и окружений, KPI – метрика #42)

3. Качество продукта

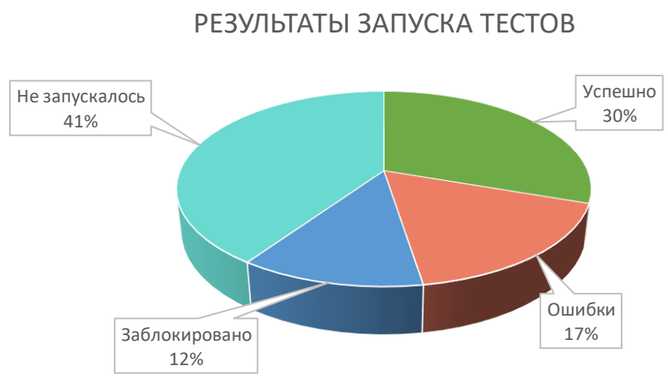
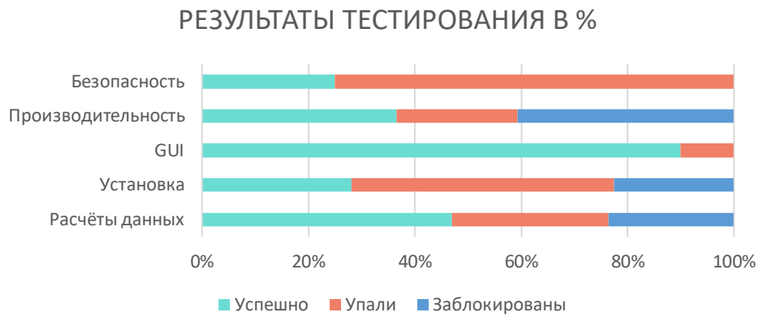
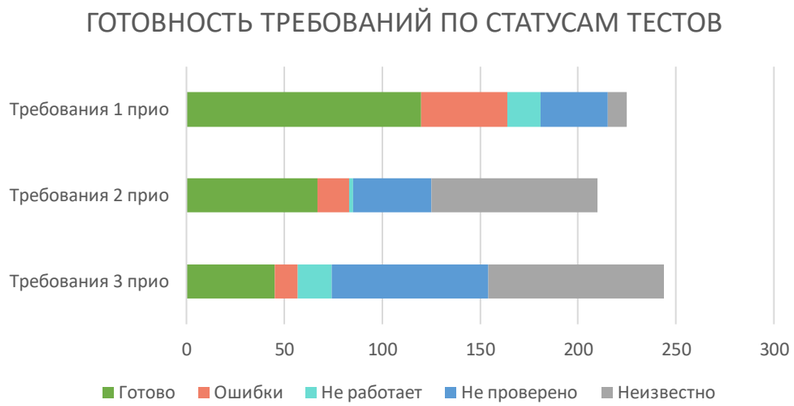
3.1 Удовлетворение пользователей

#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
43	Средняя оценка пользователей	Average Client Rating	AVG (Пользовательские оценки за период) Можно собирать в: • Прямых опросах • Маркетах Используется для оценки: • Статика по релизу • Динамика по итерациям		• KPI проекта • Выявление источников низких оценок (см. диаграмму – проблемы в iOS версии) • Отслеживание динамики внедрения изменений
44	Распределение пользовательских оценок	Clients Grades Evaluation	Сбор «сырых» данных по числу каждого типа оценки	**Оценка** | **1** | **2** | **3** | **4** | **5** **Число оценок** | 27 | 32 | 194 | 140 | 16	• Анализ исключительно высоких и исключительно низких
45	Новых запросов от пользователей за период	New Improvement Requests	• Внедрение очереди «запросы на улучшение от клиентов» в таск-трекере • Контроль статусов запросов: ○ «подвешено» ○ «в работе» ○ «реализовано» ○ «принято заказчиком»		• Оценить уровень удовлетворённости клиентов внедрением доработок • Спланировать работы по внедрению запросов от пользователей • Отфильтровать давно подвешенные доработки и принять по ним решения
46	Внедрено пользовательских запросов за период	Improvement Requests Implemented			
47	Открытых пользовательских запросов	Opened Improvement Requests			
48	Рейтинг внедрения пользовательских запросов	Improvement Ratio	$\frac{\text{Реализовано улучшений}}{\text{Запрошено улучшений}} \times 100\%$		
49	Прохождение приёмки заказчиком	Customer Acceptance Ratio	$\frac{\text{Сборок принято заказчиком}}{\text{Всего сборок за период}} \times 100\%$		• KPI проектной команды • Анализ причин непринятия сборки • Инструмент аргументации при внедрении решений по углублению тестирования

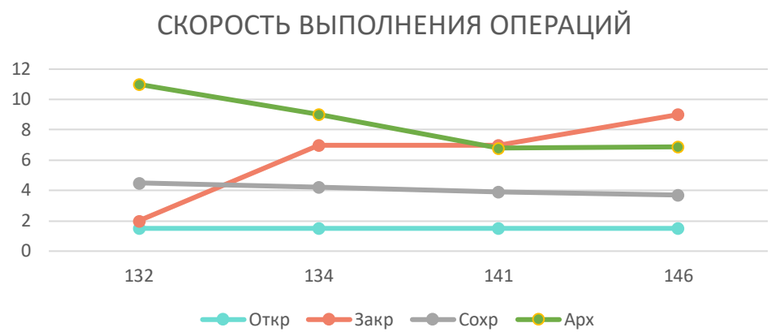
3.2 Дефекты в продукте

#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
50	Дефекты в продукте по статусам	Defects by Status	$\frac{\text{Дефектов в категории}}{\text{Всего дефектов}} \times 100\%$ Возможен сбор более комплексных метрик, например: статусы дефектов по приоритетам, критичность дефектов по областям, и т.д.		• Оценить необходимость в исправлении дефектов • Спланировать затраты на достижение требуемого уровня качества по дефектам
51	Дефекты в продукте по критичности	Defects by Severity			• Оценить приоритетность заведения багов тестировщиками • Оценить количество скрытых дефектов (при заведении только высококритичных)
52	Дефекты в продукте по области	Defects by Functional Area			• Оценить качество отдельных функциональных областей • Принять решения о расширении подкоманд разработки и/или тестирования
53	Дефекты в продукте по типу тестов	Defects by Test Type			• Оценить качество продукта по соответствию нефункциональным требованиям
54	Дефекты в продукте по платформам	Defects by Environment			• Оценить качество сборок по окружениям • Принять решение о сокращении или повышении объёмов тестирования по окружениям
55	Динамика дефектов	Defects Dynamics	Прирост дефектов = Заведено дефектов – Исправлено дефектов Число открытых дефектов = Число дефектов, открытых на начало периода, + Прирост дефектов		• Отслеживать динамику качества продукта по дефектам • Прогнозировать готовность продукта к релизу (обнаружение момента «сходимости дефектов»)

3.3 Результаты тестирования

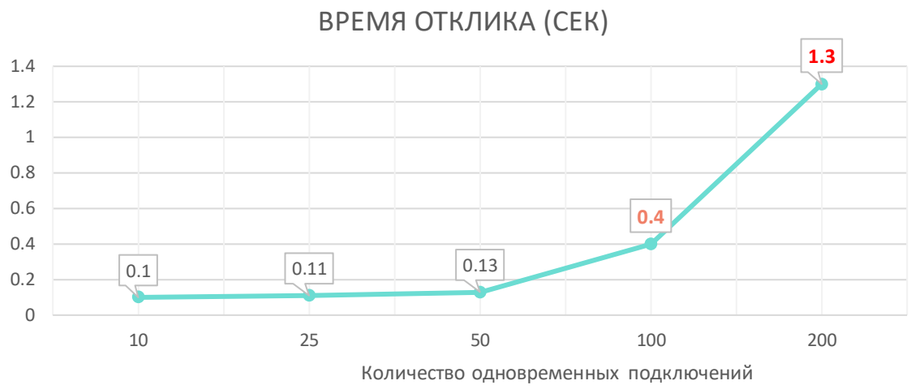
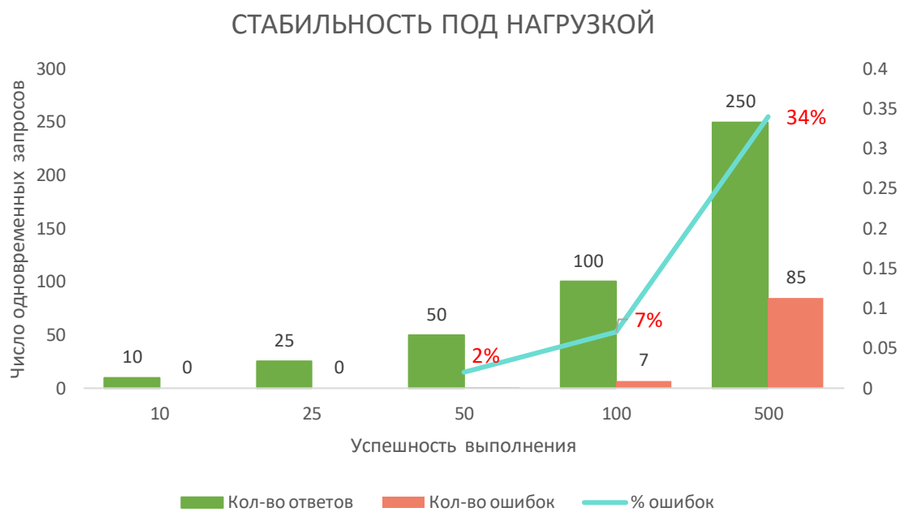
#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
56	Успешных тестов	Passed Test Cases	$\frac{\text{Пройдено тестов}}{\text{Запущено тестов}} \times 100\%$		• Показывает процент работающей функциональности (в комбинации с оценкой тестового покрытия!)
57	Упавших тестов	Failed Test Cases	$\frac{\text{Упало тестов}}{\text{Запущено тестов}} \times 100\%$		• Показывает качество и стабильность ПО
58	Заблокированных тестов	Blocked Test Cases	$\frac{\text{Заблокировано тестов}}{\text{Запущено тестов}} \times 100\%$		• Показывает объём задач, необходимых для выполнения после получения новой сборки на тестирование
59	Запущено тестов	Executed Test Cases	$\frac{\text{Запущено тестов}}{\text{Всего тестов}} \times 100\%$		• Для оценки оставшихся работ по тестированию • Для оценки достоверности данных по метрикам 56-58
60	Результаты тестирования по категориям	Test Results by Category	При наличии категорий в тестах, используется для оценки различных аспектов качества: • Типы тестов (функц, произв, нагрузка и т.д.) • По приоритетам тестов • По областям функциональности		• Оценка качества различных категорий ПО • Распределение ресурсов в наиболее проблемные области
61	Готовность требований по тестам	Requirements Readiness by Tests	1. В системе ведения требований указываются ссылки на тесты, покрывающие это требование 2. По результатам выполнения тестов проставляется статус готовности требования: • Готово - все тесты по требованию пройдены успешно • Ошибки - часть тестов по требованию упали • Не работает - все тесты по требованию упали • Не проверено - тесты по требованию не запускались • Неизвестно - к требованию не привязаны тесты		• Для оценки оставшихся объёмов тестирования • Для оценки рисков при принятии решения о релизе • Для приоритезации задач по тестированию и исправлению дефектов

3.4 Характеристики качества ПО – 1/4

<table>
<tr><th>#</th><th>Метрика рус.</th><th>Метрика англ.</th><th>Как рассчитывать</th><th>Визуализация метрики</th><th>Как использовать</th></tr>
<tr>
<td>62</td>
<td>Производительность в динамике</td>
<td>Dynamical performance</td>
<td>Оценка скорости работы основных бизнес опций (загрузка ключевых страниц, выполнение ключевых операций или запросов).
• Изменение скорости работы основных элементов приложения.
• Динамика изменения скорости работы основных элементов, относительно целевого уровня.</td>
<td>
<table>
<tr><td rowspan="2">Операция</td><td colspan="3">Сборка</td></tr>
<tr><td>7.1.132</td><td>7.1.134</td><td>7.1.141</td></tr>
<tr><td>Открытие файла</td><td>1.5</td><td>1.5</td><td>1.5</td></tr>
<tr><td>Закрытие файла</td><td>2</td><td>7</td><td>7</td></tr>
<tr><td>Сохранение файла</td><td>4.5</td><td>4.2</td><td>3.9</td></tr>
<tr><td>Архивация</td><td>11</td><td>9</td><td>6.8</td></tr>
</table>


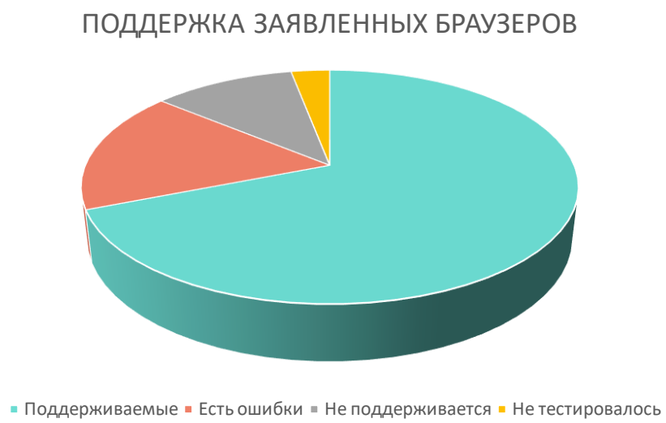
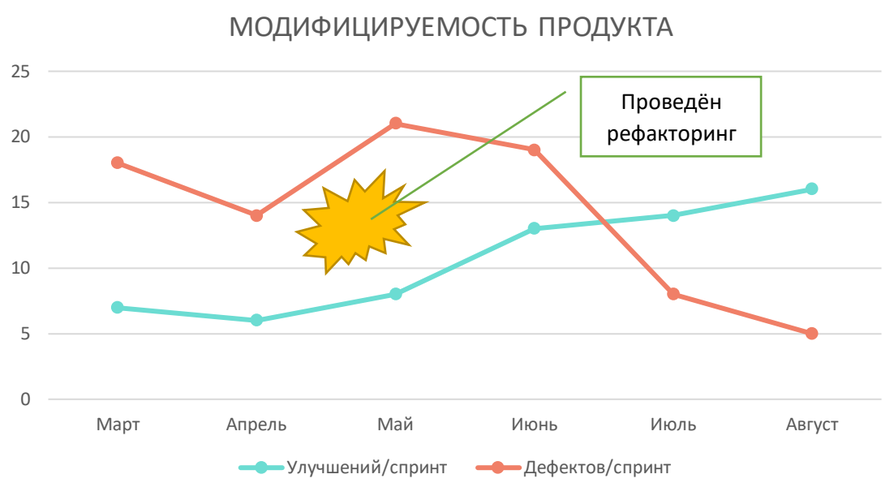
</td>
<td>• Для выявления регресса в производительности
• Для оценки улучшений
• Для оценки изменения скорости работы ключевого функционала</td>
</tr>
<tr>
<td>63</td>
<td>Производительность в сравнении с конкурентами</td>
<td>Performance compared to competitors</td>
<td>1. Оценка скорости работы основных бизнес-сценариев в сравнении с показателями конкурентов.
• По минимальному количеству действий пользователя.
• По скорости выполнения
2. Сравнение нескоростных ключевых параметров производительности, таких как
• Коэффициент сжатия при архивации
• Допускаемые пользователями ошибки
• Число одновременных подключений и т.д.</td>
<td>
<table>
<tr><th>Параметр сравнения</th><th>Наш продукт</th><th>Конкурент 1</th><th>Конкурент 2</th></tr>
<tr><td>Скорость открытия файлов, сек</td><td>2,4</td><td>2,8</td><td>6</td></tr>
<tr><td>Скорость закрытия файлов, сек</td><td>16</td><td>10</td><td>11,2</td></tr>
<tr><td>GOMS для создания отчёта</td><td>19</td><td>18</td><td>26</td></tr>
<tr><td>Размер архива с тестовыми данными, Мб</td><td>4</td><td>5</td><td>9</td></tr>
<tr><td>Процент пользовательских ошибок</td><td>1,7%</td><td>0,8%</td><td>6,4%</td></tr>
</table>
</td>
<td>• Для маркетингового продвижения
• Для обнаружения зоны развития
• Для обоснования наличия дефектов производительности
• Выявляя наилучшие значения среди конкурентов, мы определяем, к чему можно стремиться</td>
</tr>
</table>

3.4 Характеристики качества ПО – 2/4

#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
64	Производительность под нагрузкой	Performance under load	Скорость отклика при разных нагрузках, в подавляющем большинстве случаев измеряется автоматизировано (JMeter, Grinder, HP Performance Center и т.д.)		• Оценить соответствие требованиям по нагрузке • Для оценки возможности масштабирования приложения по количеству пользователей
65	Стабильность под нагрузкой	Stability under load	1. Автоматизация множества одновременных запросов к серверу 2. Сбор статистики, какой % запросов проходит успешно, и какой вызывает ошибки		• Для определения отказоустойчивости системы в стрессовых ситуациях • Соблюдение SLA (англ. Service Level Agree ment - Соглашение об уровне предоставления услуги) • Определить, на каком количестве одновременных подключений проявляются ошибки
66	Ошибки под нагрузкой	Failures during load tests	$\frac{\text{Ошибочных откликов}}{\text{Всего запросов}} \times 100\%$		

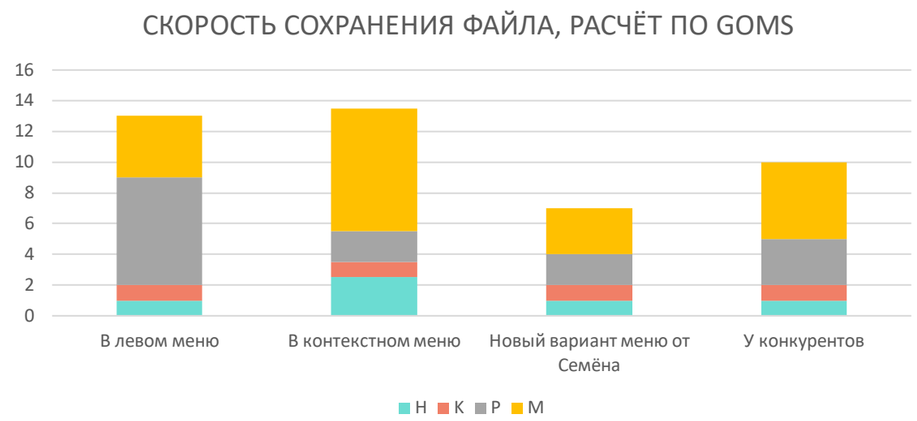
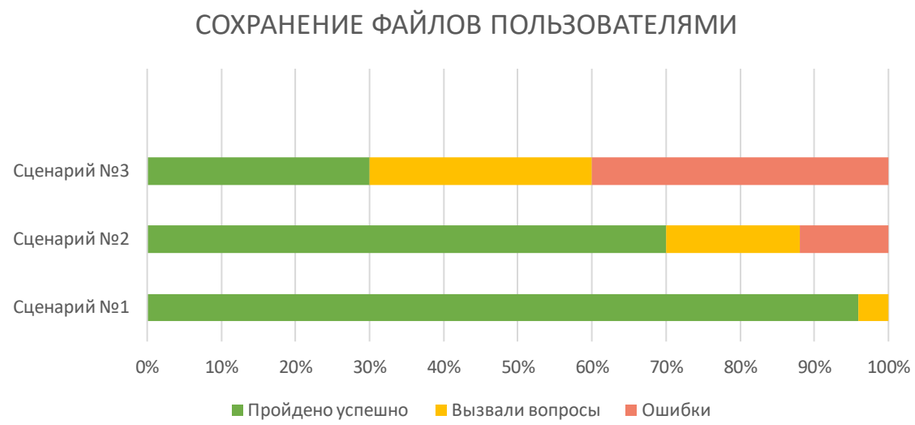
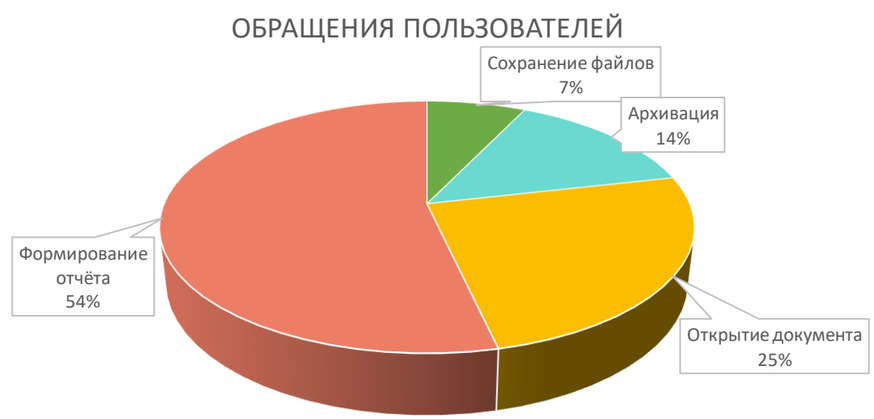
3.4 Характеристики качества ПО – 3/4

<table>
<tr><th>#</th><th>Метрика рус.</th><th>Метрика англ.</th><th>Как рассчитывать</th><th>Визуализация метрики</th><th>Как использовать</th></tr>
<tr><td>67</td><td>Совместимость</td><td>Compatibility</td><td>Процент поддерживаемых платформ, окружений, браузеров, версий ОС, разрешений экрана и т.д.
Список возможных статусов по окружению может быть разным, минимальный набор статусов:
• Поддерживается
• Не поддерживается

По каждому статусу проводится расчёт:
$\frac{\text{Статус тестирования}}{\text{Всего окружений}} \times 100\%$</td><td> 
</td><td>• Для оценки рисков по непротестированным окружениям
• Для выявления неподдерживаемых окружений</td></tr>
<tr><td>68</td><td>Модифицируемость</td><td>Modifiability</td><td>• Скорость внесения изменений по модулям продукта
• Количество обнаруженных регрессионных дефектов по итогам внесения изменений</td><td>
<table>
<tr><th>Параметр</th><th>Модули</th><th>B-1</th><th>B-2</th><th>B-3</th><th>B-4</th><th>B-5</th><th>AVG</th></tr>
<tr><td rowspan="3">Количество регрессионных дефектов</td><td>M-1</td><td>14</td><td>9</td><td>11</td><td>13</td><td>11</td><td>11.6</td></tr>
<tr><td>M-2</td><td>4</td><td>3</td><td>2</td><td>4</td><td>1</td><td>2.8</td></tr>
<tr><td>M-3</td><td>2</td><td>3</td><td>3</td><td>1</td><td>0</td><td>1.8</td></tr>
<tr><td rowspan="3">Количество улучшений в реализованный функционал</td><td>M-1</td><td>4</td><td>2</td><td>1</td><td>3</td><td>1</td><td>2.2</td></tr>
<tr><td>M-2</td><td>4</td><td>3</td><td>2</td><td>4</td><td>3</td><td>3.2</td></tr>
<tr><td>M-3</td><td>10</td><td>9</td><td>10</td><td>8</td><td>9</td><td>9.2</td></tr>
</table>


</td><td>• Выявление зон для рефакторинга
• Статистика для планирования будущих релизов и количества ожидаемых дефектов на исправление
• Оценка результативности проведения код-ревью и/или рефакторинга</td></tr>
</table>

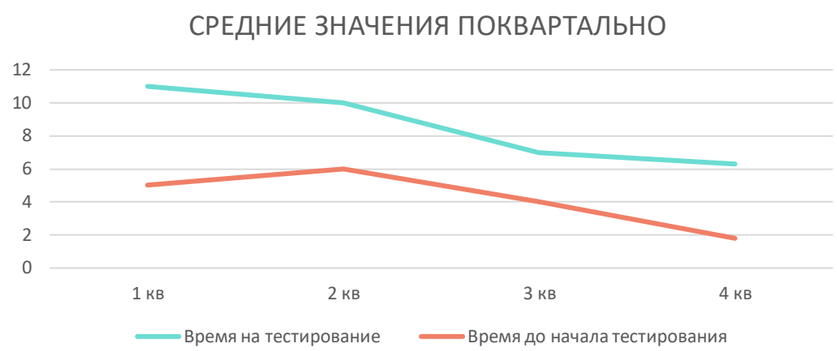
3.4 Характеристики качества ПО – 4/4

#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
69	Юзабилити отдельных форм по GOMS	Usability measured by GOMS	Модели расчетов по GOMS $t = H + K + P + M$ *где,* *t – время, затраченное на выполнение действия* *H – перенос руки на мышь* *K – нажатие клавиши клавиатуры или мыши* *P – перенос курсора* *M – обдумывание следующего шага*		• Для поиска неудачных (долгих по выполнению) реализаций в интерфейсе • Для выбора варианта реализации экранной формы • Для сравнительного анализа продукта с конкурентами
70	Юзабилити сценариев: успешность выполнения пользователями	Scenarios completion by users	• Выполнение пользователем заданных сценариев, подсчёт его ошибок и вопросов, оценка успешности выполнения сценариев • Сценарий считается успешным, если фактический результат прохождения пользователем сценария совпадает с ожидаемым $\frac{\text{Статус выполнения}}{\text{Пользователей тестировало}} \times 100\%$		• Для приоритезации юзабилити-доработок по сценариям • Для выявления непонятных мест и юзабилити-ошибок • Для обоснования необходимости исправлять ошибки
71	Структура пользовательских обращений	Users complains by product modules	Сбор статистики по обращениям в техническую поддержку в статусе «ошибка пользователя» (обращения, в которых причиной ошибки было непонимание пользователя, а не дефекты в продукте) $\frac{\text{Обращений по модулю}}{\text{Всего обращений}} \times 100\%$		• Выявить зоны, в которых необходимо проведение юзабилити-экспертизы • Обнаружить ошибки и нелогичности, которые незаметны опытным участникам проекта, привыкшим к продукту

4. Эффективность тестирования

4.1 Скорость тестирования

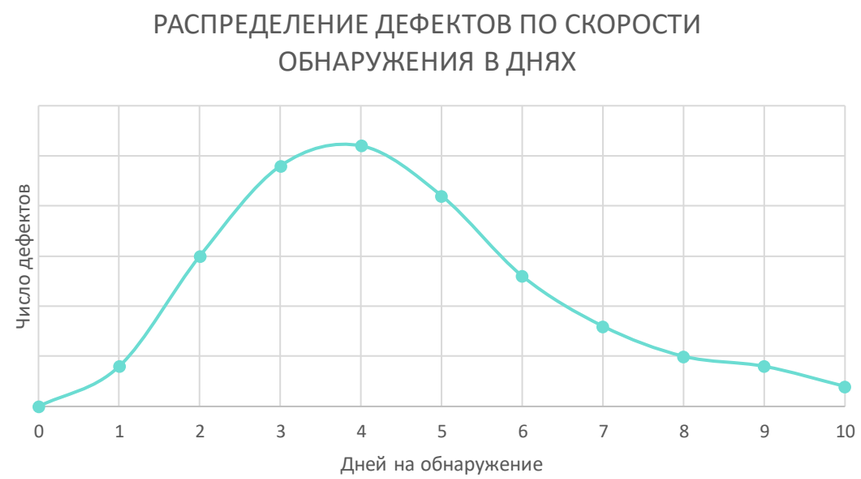
<table>
<tr><th>#</th><th>Метрика рус.</th><th>Метрика англ.</th><th>Как рассчитывать</th><th>Визуализация метрики</th><th>Как использовать</th></tr>
<tr><td>72</td><td>Время на тестирование сборки</td><td>Time to test the build</td><td>AVG (Дата завершения тестирования – Дата начала тестирования)

Измеряется в рабочих днях</td><td rowspan="2">
<table>
<tr><th>Сборка</th><th>Дата готовности</th><th>Дата начала тестирования</th><th>Дата завершения тестирования</th></tr>
<tr><td>14.1</td><td>29.11</td><td>03.12</td><td>11.12</td></tr>
<tr><td>14.2</td><td>21.12</td><td>21.12</td><td>29.12</td></tr>
<tr><td>14.3</td><td>18.01</td><td>24.01</td><td>12.02</td></tr>
<tr><td>14.4</td><td>27.02</td><td>01.03</td><td>10.03</td></tr>
</table>


</td><td rowspan="2">• Статистика для последующего планирования релизов
• Для определения зон развития / повышения скорости
• Для поиска отклонений и выявления причин (простоев и срывов сроков)
• Для оценки динамики и влияния внедрённых изменений</td></tr>
<tr><td>73</td><td>Время до начала тестирования</td><td>Time to start testing</td><td>AVG (Дата начала тестирования – Дата готовности сборки)

Измеряется в рабочих днях</td></tr>
<tr><td>74</td><td>Время на обнаружение критического дефекта</td><td>Time to start testing</td><td>AVG (Дата заведения дефекта – Дата внедрения дефекта в код)

Для внедрения этой метрики необходимо добавить в баг-трекер поле «дата внедрения дефекта». Разработчик заполняет это поле в момент смены статуса дефекта на «исправлено» (в момент исправления он уже знает, какой именно коммит вызвал этот дефект, и может посмотреть, когда он был произведён).

Так как сбор этой метрики подразумевает дополнительные затраты, обычно целесообразно собирать эту статистику только по критичным дефектам.</td><td>


<table>
<tr><th>Дефект</th><th>Дней</th><th>Причина долгого обнаружения</th><th>Решение</th></tr>
<tr><td>17382</td><td>9</td><td>Был заблокирован для обнаружения</td><td>Убрать из статистики</td></tr>
<tr><td>17315</td><td>11</td><td>Не знали о влиянии параметра в ТА</td><td>Провести ревью тестов по модулю с архитектором, возможно, ещё что-то упустили</td></tr>
</table>
</td><td>• Для выяснения средних значений (для планирования, и оценки рисков)
• Для обнаружения критичных отклонения и выяснения причин, почему так долго не могли обнаружить критичный дефект</td></tr>
</table>

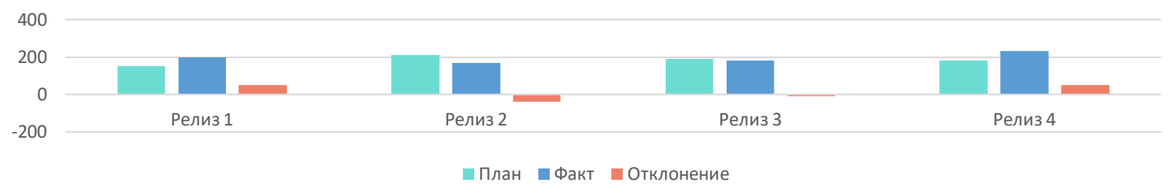
4.2 Финансовые показатели

#	Метрика рус.	Метрика	Как рассчитывать	Визуализация метрики	Как использовать
75	ФОТ (Фонд оплаты труда) в тестировании	Total Cost of Testing Labor	Фонд оплаты труда команды тестирования, включая все необходимые затраты для обеспечения деятельности сотрудников. При расчёте необходимо взаимодействие с фин. департаментом и бухгалтерией, для учёта налогов, отпусков, больничных, офисных платежей, покупки оборудования, сервисов, социальных выплат. Обычно, сумма дополнительных затрат на команду варьируется от 50 до 120% от суммы заработных плат.		• Для бюджетирования и расчётов при запросе расширения • Для контроля при необходимости сокращения
76	Затраты на тестирование	Total Cost of Testing	ФОТ команды + тестовое оборудование + тестовые окружения + используемый инструментарий + внешние подрядчики		• Для бюджетирования проекта и команды тестирования • Для обоснования расширения
77	Стоимость обнаружения ошибки	Cost per Bug Find	$\frac{\text{Затраты на тестирование}}{\text{Число дефектов}}$ Рассчитывается за период		• Для оценки экономической оправданности тестирования
78	Стоимость исправления ошибки	Cost per Bug Fix	$\frac{\text{Затраты на исправление дефектов}}{\text{Число дефектов для исправления}}$ Рассчитывается за период		• Для поиска затратных в исправлении дефектов и анализа причин
79	Отклонение бюджета тестирования	Testing Budget Variance	$\frac{\text{Факт затрат} - \text{План затрат}}{\text{План затрат}} \times 100\%$		• Для планирования бюджета в будущем • Для анализа причин отклонений

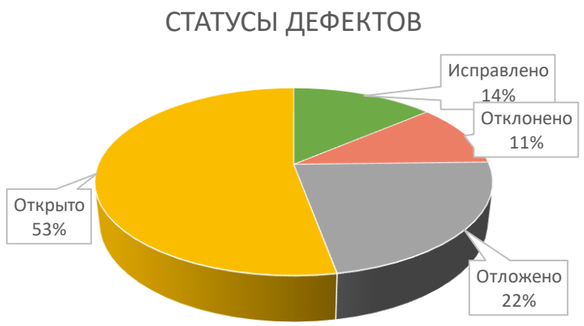
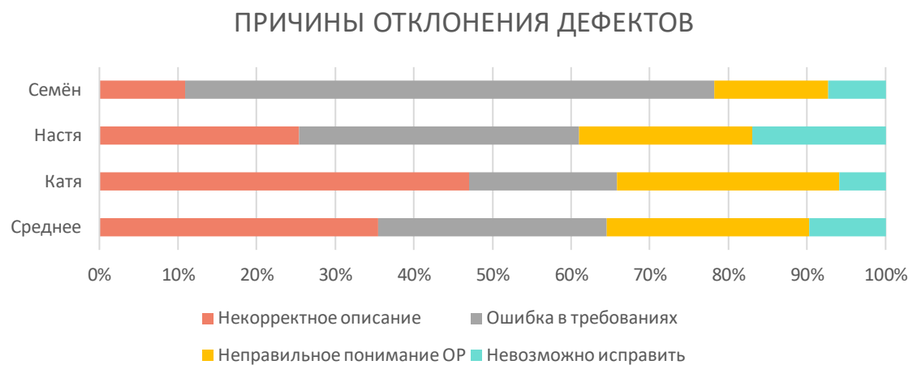
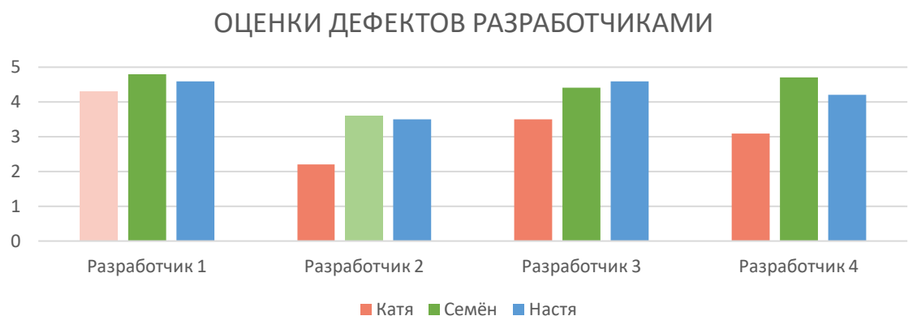
Общая стоимость тестирования

Виды расходов на сотрудника в месяц	Стоимость сотрудника 1 (руб.)	Стоимость сотрудника 2 (руб.)	Стоимость сотрудника 3 (руб.)	Всего на команду/проект (руб.)
Заработная плата(gross)	80 000	85 000	83 000	245 000
Оплата труда (net)	69 600	73 950	72 210	215 760
Налоги с з/пл.				
ПФР - 22%	17 600	18 700	18 260	54 560
ФМС - 2.9%	2 320	2 465	2 407	7 192
ФФОМС - 5.1%	4 080	4 435	4 233	12 648
НДФЛ - 13%	10 400	11 050	10 790	32 240
Отпуск сотрудника (1/12 с з/пл.)	6 667	7 083	6 917	20 667
Организационные расходы				
аренда офиса(по СанНПину на каждого сотрудника должно быть не менее 4.5 кв/м площади офиса), из расчета средней ставки аренды 1000 руб/кв.м.	4 500	4 500	4 500	13 500
ПК и оргтехника, мебель. В среднем 1/20 от з/пл. сотрудника	4 000	4 250	4 150	12 400
Административные расходы				
З/плата сотрудников HR: поиск, найм, увольнение, обучение, сопровождение *З/плата бухгалтерии: найм, увольнение, сопровождение, больничные* *З/плата руководителя (начальника) сотрудника : сопровождение, контроль, управление*	2 000	2 000	2 000	6 000
Повышение квалификации сотрудника (курсы, тренинги пр.)				
В среднем 1/12 от средней цены курса для ручного тестировщика(15 000 руб.) при условии прохождении 1-го курса в год	1 250	1 250	1 250	3 250
ИТОГО в месяц:	122 417	129 583	126 717	378 717
Стоимость команды тестирования за месяц:				378 717

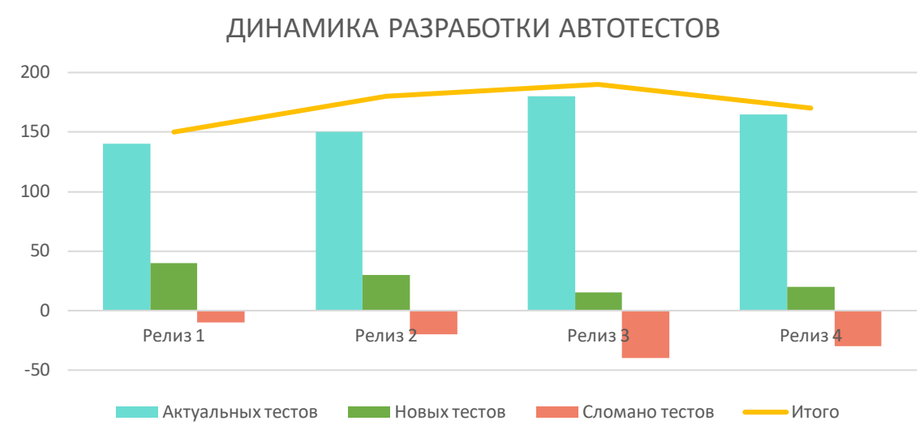
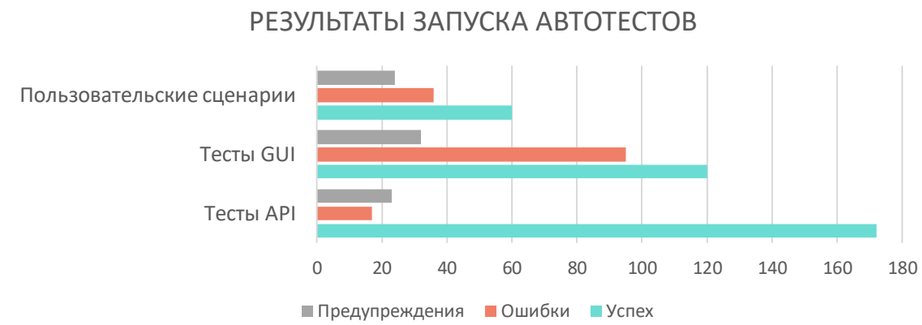
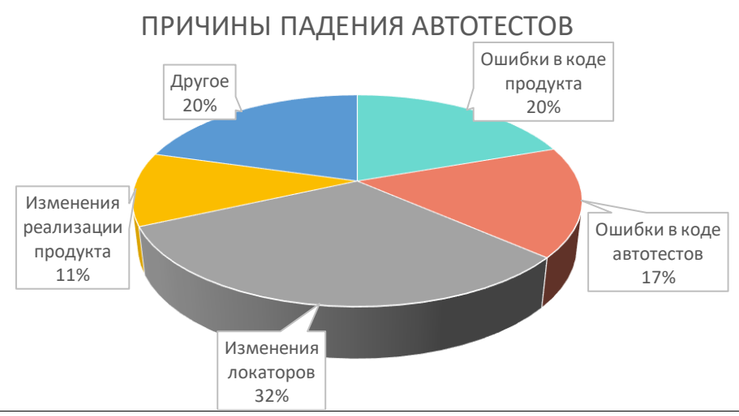
Наименование проекта	Кол-во сотрудников	Дата начала	Дата завершения	Полная стоимость тестирования (руб.)	Сумма затрат на исправление дефектов (руб.)	Общее кол-во выявленных дефектов(шт.)	Стоимость обнаружения дефекта (руб.)	Стоимость исправления дефекта
Проект 1	3	01.08.2020	30.08.2020	710 400,13	504 416,32	158	4 476,30	4 097,70
Проект 2	3	01.08.2020	30.08.2020	563 109,62	914 216,19	321	1 754,23	4 932,64
Проект 3	3	01.08.2020	30.08.2020	314 043,90	92 116,70	38	8 264,31	2 791,12



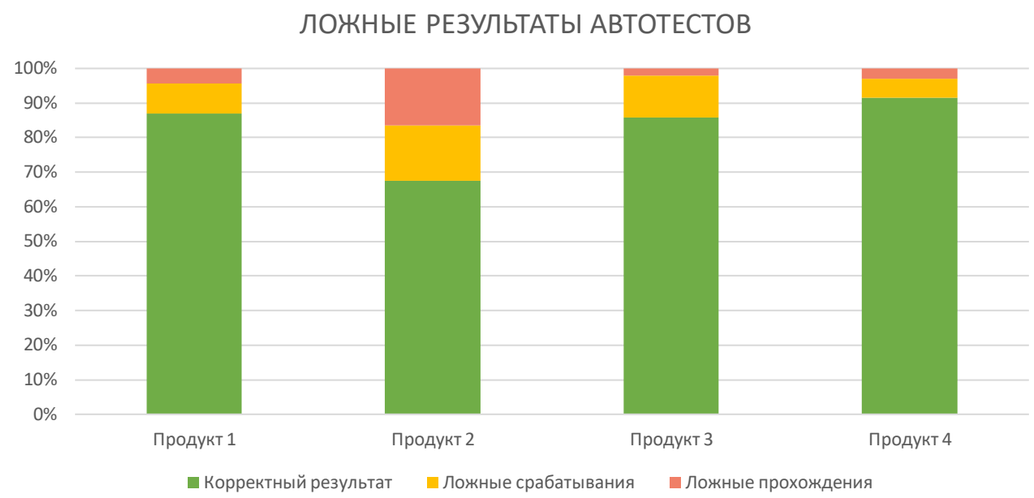
4.3 Работа с дефектами

#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
80	% исправленных дефектов	Bugs Fix Ratio	$\frac{\text{Исправлено дефектов}}{\text{Заведено дефектов}} \times 100\%$ Расчёт проводится за период или по версии/итерации		• Для выявления причин НЕисправления дефектов. • Для оценки статуса тестирования по модулям, направлениям, типам тестов
81	% отклонённых дефектов	Bugs Reject Ratio	$\frac{\text{Отклонено дефектов}}{\text{Заведено дефектов}} \times 100\%$ Расчёт проводится за период или по версии/итерации		
82	Причины отклонения дефектов по группам	Rejection Causes	Для сбора этой метрики требуется выпадающий список в баг-трекере, необходимый для заполнения разработчиком при отклонении дефекта, например: • не воспроизводится • неправильное понимание ожидаемого результата • исправление невозможно • ошибка в требованиях $\frac{\text{Причина отклонения}}{\text{Всего отклонено}} \times 100\%$		• Чтобы найти способ сократить число отклонённых дефектов (решить проблему, вызывающую наибольшее число отклонений)
83	Качество заведения дефектов	Bugs Submission Quality	Субъективная оценка в баг-трекере, в разрезе разработчиков, тестировщиков, типов дефектов и т.д.		• Оценить качество заведения дефектов на проекте, в команде, у отдельного сотрудника • Выяснить причины анормально низких оценок для решения • Учитывать разработчиков, которые могут завышать или занижать оценки

4.4 Автоматизированное тестирование – 1/2

#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
84	Скорость прохождения автотестов	Autotests execution speed	Время прохождения автотестов: $AVG\ (\text{Время готовности отчёта по автотестам} - \text{Время запуска автотестов})$ Среднее время на выполнение 1 теста: $\frac{\text{Время прохождения автотестов}}{\text{Число автотестов}}$		• Для планирования тестовых циклов • Для принятия решения об эффективности автотестов и их расширении • Для обнаружения проблем со скоростью автотестов • Для своевременного обнаружения регресса производительности
85	Скорость разработки автотестов	Autotests implementation speed	$\frac{\text{Затраты на автоматизацию за период}}{\text{Число новых автотестов за период}}$ Может считаться в разрезе сотрудников, команд, типов автотестов и т.д.		
86	Стабильность автотестов	Autotests Stability	$\frac{\text{Успешно пройдено автотестов}}{\text{Всего автотестов}} \times 100\%$ Измеряется как среднее за период или релиз Может группироваться по разработчику автотестов, функциональной области продукта, и т.д.		• Для планирования трудозатрат на поддержку и анализ автотестов • Для оценочного сравнения стабильности автотестов по категориям (разработчик, тип тестов, функц. область и т.д.)
87	Причины нестабильности тестов по категориям	Autotests Failures Causes	Заполнение из выпадающего списка каждый раз, когда вносятся изменения в автотесты, для фиксации причины изменений, и дальнейший расчёт доли каждой причины: $\frac{\text{Причина падения автотестов}}{\text{Всего падений автотестов}} \times 100\%$		• Для поиска ключевой причины нестабильности автотестов • Для решения проблемы, вызывающей наибольшее число падений • В редких случаях – для принятия решения о смене стратегии автоматизации (выбор других интерфейсов, инструментов, отказ от автоматизации новой функциональности и т.д.)

Релиз	Время на автотестирование сборки	Время на прохождение 1 теста	Затраты на поддержку 1 автотеста	Затраты на разработку 1 автотеста
Релиз 1	9 ч	7 мин	0,2 ч/ч	3,4 ч/ч
Релиз 2	9 ч	6,5 мин	0,2 ч/ч	4,2 ч/ч
Релиз 3	9,3 ч	7,2 мин	0,2 ч/ч	17 ч/ч
Релиз 4	9,5 ч	7,6 мин	0,3 ч/ч	6 ч/ч

#	Метрика рус.	Метрика англ.	Как рассчитывать	Визуализация метрики	Как использовать
88	Ложные срабатывания	Autotesting False Alarms	При анализе причин падения автотестов выявлять все результаты, которые не были вызваны ошибками в продукте		• Для оценки уровня достоверности автотестов • Для оценки возможности принятия решений о релизе продукта по итогам автотестов (насколько достоверны показатели автотестов до проведения ручного анализа)
89	Ложные прохождения тестов	Autotesting False Positives	При регистрации дефектов в областях, покрытых автотестами, уточнять, почему дефект не был зарегистрирован по итогам запуска автотестов. Если обнаруживаются автотесты, которые пропускают ошибки (нехватка проверок, ошибки в логике), то такие ситуации помечаются как ложные прохождения тестов. **!Важно** Ложные прохождения, как и пропуски в ручном тестировании, невозможно выявить **все**, и это будет примерная оценка.		
90	Экономическая оправданность автотестов	Autotests Return of Investment	$\frac{\text{Выгода от внедрения автотестов}}{\text{Затраты на внедрение автотестов}} \times 100\%$ Где выгода от внедрения учитывает все затраты на ручное тестирование, а затраты на автоматизацию включают в себя • Разработку автотестов • Поддержку и актуализацию автотестов • Запуски и анализ результатов автоматизированного тестирования	**Calculation table** [1]	• Для принятия решений о внедрении автоматизированного тестирования • Для выбора и приоритезации автотестов исходя из их экономической оправданности
91	Покрытие автотестами функционала продукта	Features Coverage by Autotests	$\frac{\text{Функций тестироуется автоматизировано}}{\text{Всего функций в продукте}} \times 100\%$ В качестве функций могут выступать функциональные требования, пользовательские сценарии, области и модули продукта и т.д.		• Для контроля следования плану • Для оценки возможности сокращения ручного тестирования • Для поиска зон, в которых необходимо расширять автоматизацию тестов
92	Покрытие автотестами кода продукта	Code Coverage by Autotests	См. метрики #9-12		
93	Автоматизировано регрессионных тестов	Regression Coverage by Coverage by Autotests	$\frac{\text{Автоматизировано регр.} - \text{x тестов}}{\text{Всего регрессионных тестов}} \times 100\%$		
94	Выполнение плана автоматизации тестирования	Automation Plan Fulfilment	$\frac{\text{Автоматизировано тестов}}{\text{План автоматизировать тестов}} \times 100\%$		

[1]

Calculation table

#	Scenario Name	TC's amount	Priority	Automation, man*days	1 TC automation, man*hrs	Manual testing, man-days	Results invest-n, man*hrs	Executio n times	Automation efficacy	Saved time, man-days	Reusable ?	Decision
1	OAS BVT	10	High	4	2	0.5	1	100	263%	31		
2	OAS common functionality	50	High	4	2	3	3	20	250%	36		
3	OAS iswift, ichecker	12	Low	2	2	1	2	10	133%	2.5		
4	OAS reporting	12	Medium	8	2	0.5	2	20	63%	-6		

Agenda

Field	Description
Scenario Name	Название сценария, содержащего в себе однотипные тест-кейзы.
TC's amount	Количество кейзов в сценарии.
Priority	Приоритет автоматизации для тест-менеджера. Информационное поле.
Automation, man*days	Ресурсозатраты на автоматизацию сценария (без учёта тестовых данных для каждого тест-кейза)
1 TC automation, man*hrs	Затраты на автоматизацию одного тест-кейза
Manual testing, man-days	Затраты на ручное тестирование сценария в целом
Results investigation, man*hrs	Затраты на исследование результатов прогона автотеста
Execution times	Количество ТРЕБУЕМЫХ прогонов тестов за период работы над проектом. Данное число отражает ожидаемое количество прогонов с учётом стабильности функционала, требуемого графика прогонов тестов (регулярности получения информации) и т.д.
Automation efficacy	Результирующая эффективность автоматизации тестирования в % сэкономленного времени. Считается автоматически. Учитывая погрешности вычислений, эффективной можно признать автоматизацию с показателем более 150%
Saved time	Количество человеко-дней, сэкономленных за время всего проекта. Считается автоматически

Авторы документа с непередаваемым восхищением приветствуют тех, кто осилил всё написанное, и дочитал этот документ до конца. Мы очень надеемся, что информация, изложенная в нём, была полезной и применимой.

Если у вас остались вопросы по внедрению и использованию метрик, рассмотренных в этом документе или вы не нашли в этом списке метрики, которая вам нужна для оценки рабочей деятельности.
Свяжитесь с нами.
Мы обязательно поможем решить эти задачи!

Если вы хотите знать насколько эффективной может быть ваша команда при грамотно организованных процессах и правильно подобранных инструментах - то будем рады провести **QA-аудит!**

от 3 недель и от 60 000 ₽

Хочу бесплатную консультацию