

1. Завершение тестирования



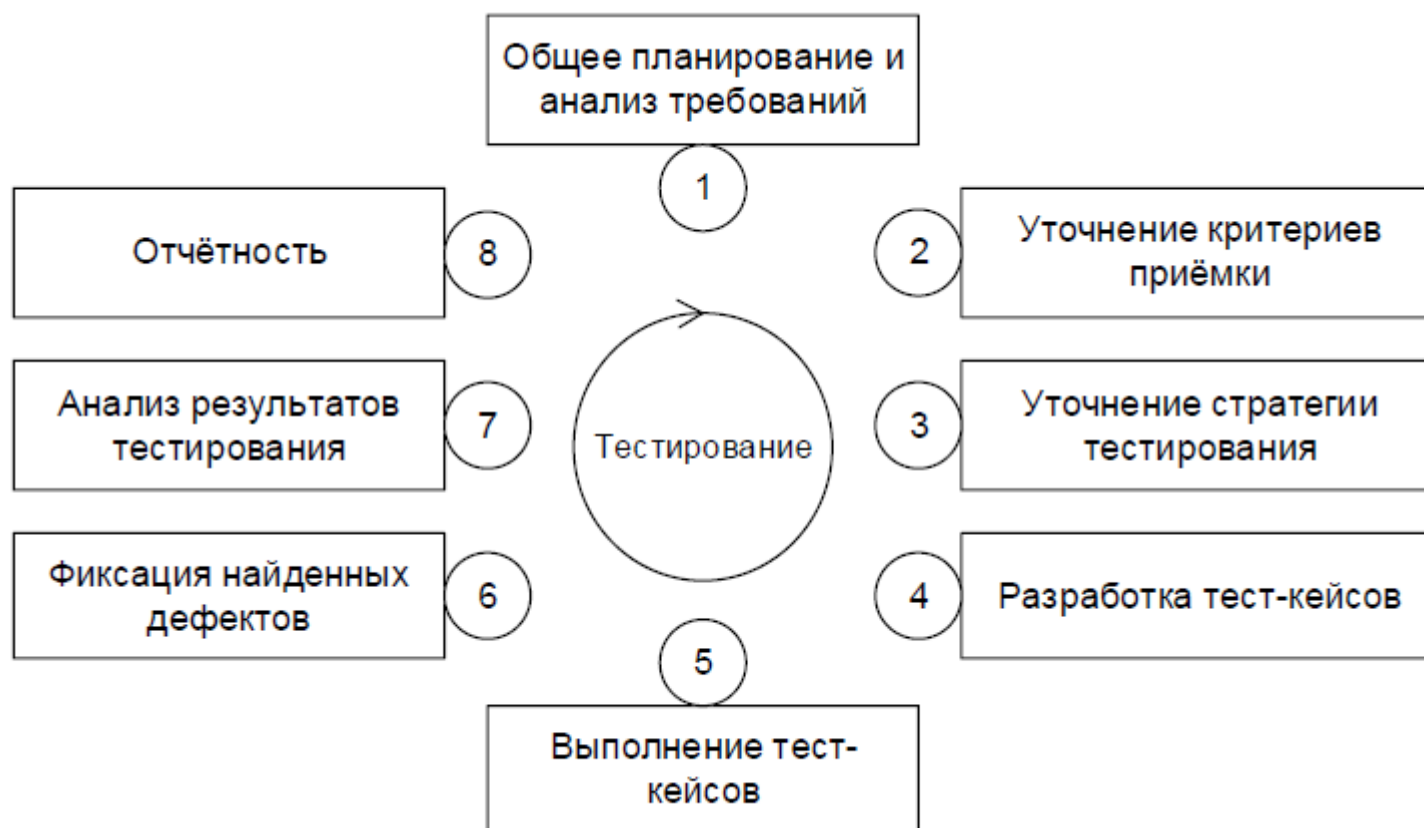
2. Документирование результатов тестирования

Повторяем:

1. Что такое тестирование?
2. Какие артефакты тестирования готовят тестировщики.
3. Описать каждый артефакт тестирования.
4. Описать основные этапы тестирования.

Жизненный цикл тестирования

Жизненный цикл тестирования обычно выражается замкнутой последовательностью действий.



Основные сложности тестирования

1. ПО по своей природе является **концептуальным**. В отличие от, например, здания или любого другого физического объекта, сложно «посмотреть» на программный продукт и оценить степень его завершённости.

2. Недостаток контроля. Поскольку ПО является нематериальным в физическом смысле, его сложно контролировать. Без точной оценки процесса разработки срываются графики выполнения работ и превышаются установленные бюджеты. Очень сложно оценить объём выполненной и оставшейся работы.

3. Отсутствие связи между отдельными событиями проекта приводит к его провалу. Ценность **прослеживаемости** особенно высока в ситуациях, когда в одной из версий продукта возникает проблема, которая оказывает влияние на другие версии этого продукта или на другие продукты. Отсутствие связи между событиями проекта приводит к тому, что решение одной проблемы увеличивает проблему в другой области или приводит к неудаче в попытке решить аналогичную проблему где-то в другом месте.

4. Недостаток мониторинга. Невозможно выполнить мониторинг проекта, если у менеджера проекта нет инструментальных средств, чтобы следить за фактической разработкой продукта в пределах проекта. В такой ситуации руководство не может принимать компетентные решения, поэтому графики продолжают срываться, а затраты продолжают превышать установленный бюджет.

Основные сложности тестирования

5. Неконтролируемые изменения. ПО является достаточно гибким, оно представляет результат работы большого коллектива, но у потребителей постоянно возникают новые идеи относительно данного программного продукта. В крупных проектах рекомендуется использовать инструментальные средства, позволяющие гибко управлять изменениями в проекте.

6. Групповой синдром разработчика. Если для разработки проекта требуется более одного разработчика, то возникает проблема группы людей, работающих над одной «базой продукта».

Усилия тратятся впустую, если несколько человек работают над одним и тем же файлом, а затем его сохраняют – сохраняются только изменения, записанные тем, кто работал над файлом последним. Самое простое решение этой проблемы лежит в блокировании файла на время работы с ним одним из сотрудников, чтобы предотвратить одновременную работу над ним нескольких человек.

7. Множественность версий. Совершенствование базового продукта приводит к выпуску дополнительных версий с самыми последними изменениями. Если в программе обнаружены ошибки, изменения необходимо сделать во всех версиях. Как только в продукте появляются новые свойства, они должны быть доступны для всех пользователей независимо от времени выпуска версии продукта.

Основные сложности тестирования

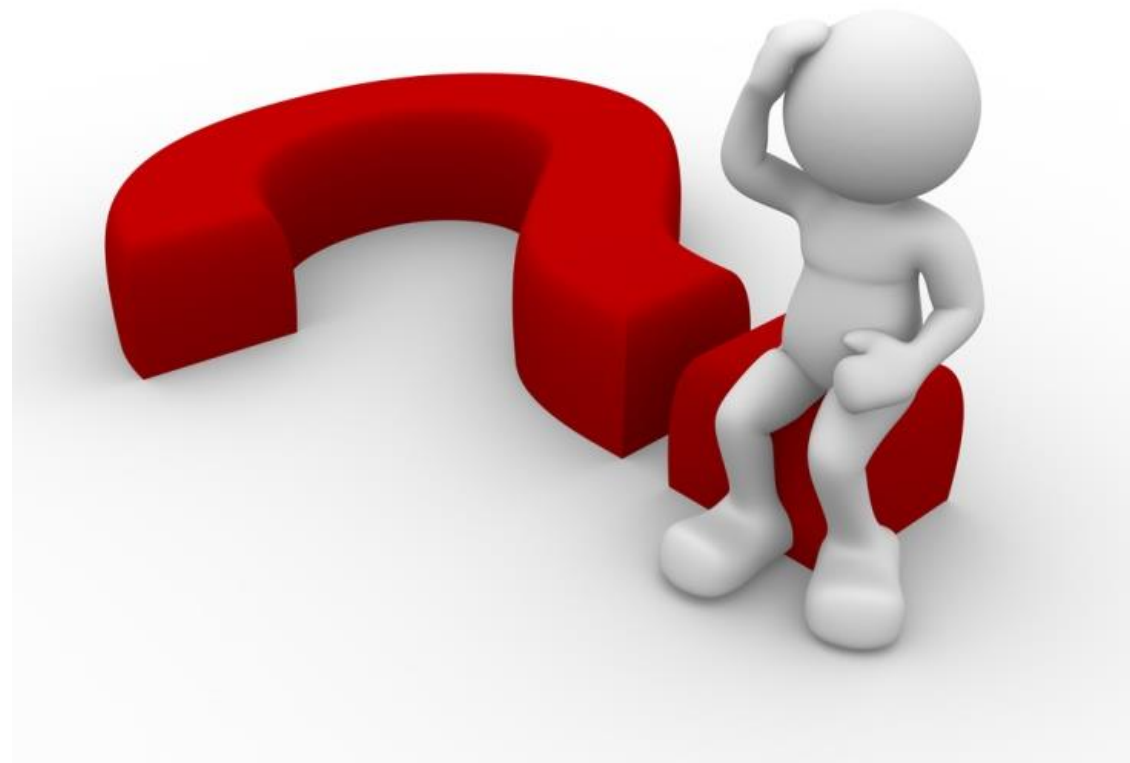
8. Семейство программных продуктов. Если программный продукт работает с четырьмя версиями Windows, тремя версиями Unix, Linux и др. платформами, то для этих платформ необходимы разные процессы установки ПО, настройки, особенности реализации и т.д.

9. Изменение графика работ. Поскольку технические требования меняются в процессе работы с проектом, должен меняться и график их выполнения.

10. Изменения штата сотрудников.

11. Изменения во внешней среде. Некоторые проекты разрабатываются на протяжении достаточно длительного времени (история знает проекты, разрабатываемые более 15-ти лет). За это время меняются не только требования к проекту, меняется очень многое во всей IT-индустрии. Выходят новые версии ОС, появляется новое аппаратное обеспечение, появляются новые технологии. Для успешного выполнения проекта такие изменения следует учитывать ещё на стадии планирования.

Когда прекращать тестирование?



Завершение тестирования

Как говорил Эдгар Дейкстра (1970):

«Тестирование программ может использоваться для демонстрации наличия ошибок, но оно никогда не покажет их отсутствие».

Программа считается готовой к выпуску, когда устранены абсолютно все критические ошибки и ~85 % не критических ошибок. Считается, что дальнейшее тестирование экономически не целесообразно.

Так как у программного обеспечения (ПО) отсутствует эталон, к которому необходимо стремиться, чтобы оно считалось полностью протестированным, то следует стремиться к некоторым уровням качества:

- ✓ отсутствие остановок ПО
- ✓ отсутствие синтаксических ошибок
- ✓ выполнение функций ПО описанных в ТЗ без ошибок

Эвристики

Эвристики — это быстрые, недорогие *способы решения проблемы или принятия решения.*

Эвристики подвержены ошибкам, то есть они могут, как сработать, так и не сработать.

Таких эвристик на данный момент существует 11 штук.

Каждая из них работает в определенном контексте, поэтому предполагается, что они будут использоваться людьми, имеющими знания и навыки для их разумного использования.

Эвристики

Эвристика «Время вышло!».

Для многих специалистов по тестированию это наиболее распространенная эвристика: тестирование останавливается, когда заканчивается выделенное на него время.

Эвристики

Эвристика «мертвой лошади» (*The Dead Horse Heuristic*).

В программе слишком много ошибок, так что продолжение тестирования не имеет смысла. Мы знаем, что все изменится настолько, что сведет на нет результаты текущего тестирования.

Эвристики

Эвристика «освежающей паузы» (*The Pause That Refreshes Heuristic*).

Предполагает приостановку тестирования на некоторое время. Можно остановить тестирование и сделать перерыв, когда устали, когда стало скучно или пропало вдохновение. Можно сделать паузу на то, чтобы выполнить некоторые исследования, разработать планы, поразмыслить над тем, что делали в прошлом и понять, что делать дальше.

Идея заключается в том, что требуется определенный перерыв, после которого требуется вернуться к продукту со свежим взглядом или свежими мыслями.

Эвристики

Эвристика «Отсутствие продвижения» (*The Flatline Heuristic*).

Что бы ни делали, получаем один и тот же результат.

Это происходит в случае, когда программа падает определенным способом или перестает отвечать, но также продвижение отсутствует, когда программа в основном ведет себя стабильно: "выглядит хорошо! "

Эвристики

Эвристика «Больше нет интересных вопросов» (*No more interesting questions*).

В этот момент решаем, что не осталось вопросов, ответы на которые были бы достаточно ценными, чтобы оправдать стоимость продолжения тестирования, и поэтому останавливаемся.

Эта эвристика используется в основном как дополнение к другим эвристикам, помогая принять решение о том, есть ли какие-то вопросы или риски, которые отменяют действие этих эвристик.

Кроме того, если одна эвристика советует нам прекратить тестирование, следует проверить, нет ли интересных вопросов или серьезных рисков в других областях, и если они есть, то скорее продолжим тестирование, чем остановимся.

Эвристики

Эвристика «Пиньяты» (*The Piñata Heuristic*).

Тестирование прекращается в тот момент, когда возникает достаточно явная серьезная проблема.

«Прекращаем ломать программу, когда начинают выпадать конфеты» – останавливаем тестирование, когда видим первую достаточно серьезную проблему.

Эвристики

- Эвристика «Задание выполнено» (*The Mission Accomplished Heuristic*).

Тестирование останавливается, когда найдены ответы на все поставленные вопросы.

Эвристики

Эвристика «Отмена задания» (*The Mission Accomplished Heuristic*).

Тестирование прекращается тогда, когда это требует заказчик.

Это может произойти по причине перерасхода бюджета, или вследствие отмены проекта, и по любой другой причине. Какова бы ни была причина, необходимо остановить тестирование.

Эвристики

Эвристика «Зашел в тупик» (*The I Feel Stuck! Heuristic*).

По какой бы то ни было причине тестирование останавливается, поскольку обнаруживается некое препятствие:

- Нет информации, которая требуется (например, многие люди заявляют, что не могут тестировать без достаточного количества спецификаций).
- Имеется блокирующая ошибка, и таким образом не можем перейти в ту область продукта, которую необходимо протестировать
- Нет необходимого оборудования или инструментария
- У команды нет квалификации, требуемой для выполнения некоторых специальных тестов.

Эвристики

Эвристика «Привычного завершения» (*The Customary Conclusion Heuristic*).

Останавливается тестирование тогда, когда обычно останавливаем тестирование.

Имеется протокол, задающий определенное количество идей для тестирования, или тест-кейсов, или циклов тестирования, или как вариант – имеется определенный объем работ по тестированию, который выполняется и после останавливаемся.

Эвристики

Эвристика «Уклонения/безразличия» (*The Avoidance/Indifference Heuristic*).

Такой вариант возможен в том случае, если тестировщикам не интересно как работает программа, или тестируемое ПО является первой версией, которую вскоре заменят.

Некоторые люди прекращают тестирование по причине лени, злого умысла или отсутствия мотивации.

Иногда бизнес-критичность выпуска нового релиза настолько высока, что никакая мыслимая проблема не остановит выход программы, и поэтому никакие новые результаты тестирования не будут иметь значения.

Подводя итоги

Тестирование программного обеспечения — это трудоемкий процесс, который не подвергается определенным правилам и не всегда следует по определенному алгоритму.

Нельзя сказать, что для определенного типа программ больше подходит определенная технология тестирования.

Для каждого определенного случая подбирается своя методика тестирования и свои эвристики остановки.

Этот выбор зависит от многих критериев, таких как тип ПО, цели создания ПО, задачи, для выполнения которых создавалось данное ПО.

Отчёт о результатах тестирования

- Анализ результатов тестирования проводится с некоторой периодичностью в процессе работы с проектом, а также в конце работы с проектом.
- *Его основная задача* – оценить текущее или финальное качество проекта и принять (если необходимо) – соответствующие решения и меры.

Отчёт о результатах тестирования

Отчёт о результатах тестирования (*test result report, TRR*) – часть тестовой документации, включающая в себя описание процесса тестирования, суммарную информацию о протестированных за подотчётный период билдах, информацию о деятельности тестировщиков, а также некоторые статистические данные.

Цель написания отчёта о результатах тестирования

Цель написания TRR – предоставление лицам, заинтересованным в проекте, полной и объективной информации о текущем состоянии качества проекта. Эта информация выражается в конкретных фактах и цифрах.

Отчёт о результатах тестирования

- Обычно, TRR предоставляется для ознакомления всей проектной команде и заказчику. Опытный менеджер проектов всегда собирает максимально полную информацию о состоянии дел, прежде чем сделать те или иные выводы.
 - ❑ Он не станет опираться только на информацию, предоставленную отделом разработчиков, т.к. разработчики часто склонны давать более оптимистичную оценку проекта, чем есть на самом деле.
 - ❑ В то же время тестировщики, во-первых, не заинтересованы в приукрашивании отчётов, а во-вторых – обладают более полной информацией о текущем состоянии качества продукта, чем какая бы то ни было другая часть проектной команды.

Периодичность выпуска отчёта о результатах тестирования

- TRR создаётся, как правило, по некоторому расписанию. Например – раз в неделю. Однако, частота выпуска отчётов может быть иной, если того требует специфика проекта (например, раз в две недели, или дважды в неделю и т.п.)
- Вместо календарного графика выпуск отчётов может быть привязан к завершению тестирования очередного билда приложения. За выпуск TRR отвечает лидер команды тестировщиков. TRR, как правило, обсуждается на еженедельных собраниях команды тестировщиков. Иногда на такие собрания приглашаются также представители других проектных команд, руководство, представители заказчика. Такие расширенные собрания особенно нужны, если на проекте возникают проблемы, требующие пересмотра стратегии разработки и тестирования проекта, сроков, финансовых вопросов и т.п.
- TRR создаётся на основе принятого в компании или предоставленного заказчиком шаблона.

Структура отчёта о результатах тестирования

1. Команда тестировщиков
2. Описание процесса тестирования (*testing process description*)
3. Краткое описание (*summary*)
4. Расписание (*testing timetable*)
5. Рекомендации (*recomendations*)
6. Статистика по ошибкам (*bugs statistics*)
7. Список новых ошибок (*new bugs found*)
8. Статистика по всем ошибкам (*all bugs statistics*)

Команда тестировщиков

- В этой части TRR перечисляются все задействованные в процессе тестирования сотрудники с указанием занимаемой должности и роли на проекте в подотчётный период.

ФИО	Должность	Роль в подотчетный период
-----	-----------	---------------------------

Описание процесса тестирования

(testing process description)

- В этой части TRR даётся краткое описание того, как происходило тестирование: какие использовались методы, техники, инструментальные средства и т.п.

- Пример:

Приложение было протестировано под ОС Windows XP sp2.en с использованием браузера FireFox 3.0.

Смоук-тест был выполнен с использованием средства автоматизации JUnit 4.0.

Тест критического пути и расширенный тест были выполнены вручную согласно документу «Тесты для ручного тестирования VWS версия 3.4.76.doc».

Подробная информация о стратегии тестирования представлена в документе «Стратегия тестирования VWS версия 5.78.23.doc».

Краткое описание (*summary*)

- В этой части TRR даётся краткое описание того, какие билды были протестированы, есть ли в качестве приложения прогресс или регресс, есть ли какие-либо проблемы, требующие внимания руководства.
- Краткое описание – важная часть отчёта, т.к. менеджеру проекта приходится просматривать огромное количество документации, и он часто принимает решение о необходимости более детального изучения отчёта как раз на основе краткого описания.

Краткое описание (summary).

Пример

Билд 1.0.78 был успешно установлен под обеими платформами (Windows XP sp2.en, FedoraCore 6.0). Смоук-тест пройден успешно. Приложение работает стабильно, основная функциональность работоспособна. Существующие проблемы в основном связаны с функциональностью, реализованной с момента выпуска последнего билда. Большинство найденных ранее ошибок успешно устранено и верифицировано. Наблюдается значительный прогресс в качестве приложения. На текущий момент выполнено более 80% запланированных тестов (остальные планируется выполнить до конца следующей недели). Было обнаружено всего четыре новых ошибки с важностью «высокая».

Расписание (*testing timetable*)

- В данном разделе отчёта приводится детализированное описание того, какая работа и на протяжении какого времени выполнялась каждым тестирующим.

ФИО	Дата	Описание	Длительность, ч
-----	------	----------	-----------------

Рекомендации (*recomendations*)

- В этой части TRR следует подчеркнуть те важные моменты, на которые следует обратить внимание руководству или лидерам проектных команд. Здесь также, возможно, будет дана рекомендация на передачу проекта заказчику («передачу в продакшн»).

Примеры

1. Билд 3.45 рекомендован в продакшн. Билд работает стабильно. Все найденные баги закрыты.
2. Рекомендуется уделить особое внимание регрессионному тестированию в связи с резким возрастанием количества багов, найденных в ранее реализованной и протестированной функциональности.
3. В связи с возникшими сложностями по тестированию приложения под FreeBSD рекомендуется рассмотреть возможность подключения к проекту специалистов по тестированию под данной платформой.

Статистика по ошибкам (*bugs statistics*)

- Здесь приводится сводная таблица, содержащая информацию об ошибках, с которыми команде тестировщиков приходилось иметь дело в *подотчётный период*.

Статус	Количество	Важность			
		Критическая	Высокая	Средняя	Минимальная
Исправлено	28	3	7	10	8
Проверено	17	5	2	5	5
Открыто заново	2	0	1	1	0
Найдено	36	5	11	8	12
Отклонено	4	0	0	1	3

Список новых ошибок (*new bugs found*)

- Здесь приводится список ошибок, обнаруженных командой тестировщиков за *подотчётный период*.
- Список ошибок легко извлечь из баг-трекинговой системы.

Идентификатор	Важность	Описание
VWS10070800723	Высокая	При входе на форум гость получает права пользователя
VWS10070800724	Критическая	СУБД зависает при достижении БД объема 400 Mb

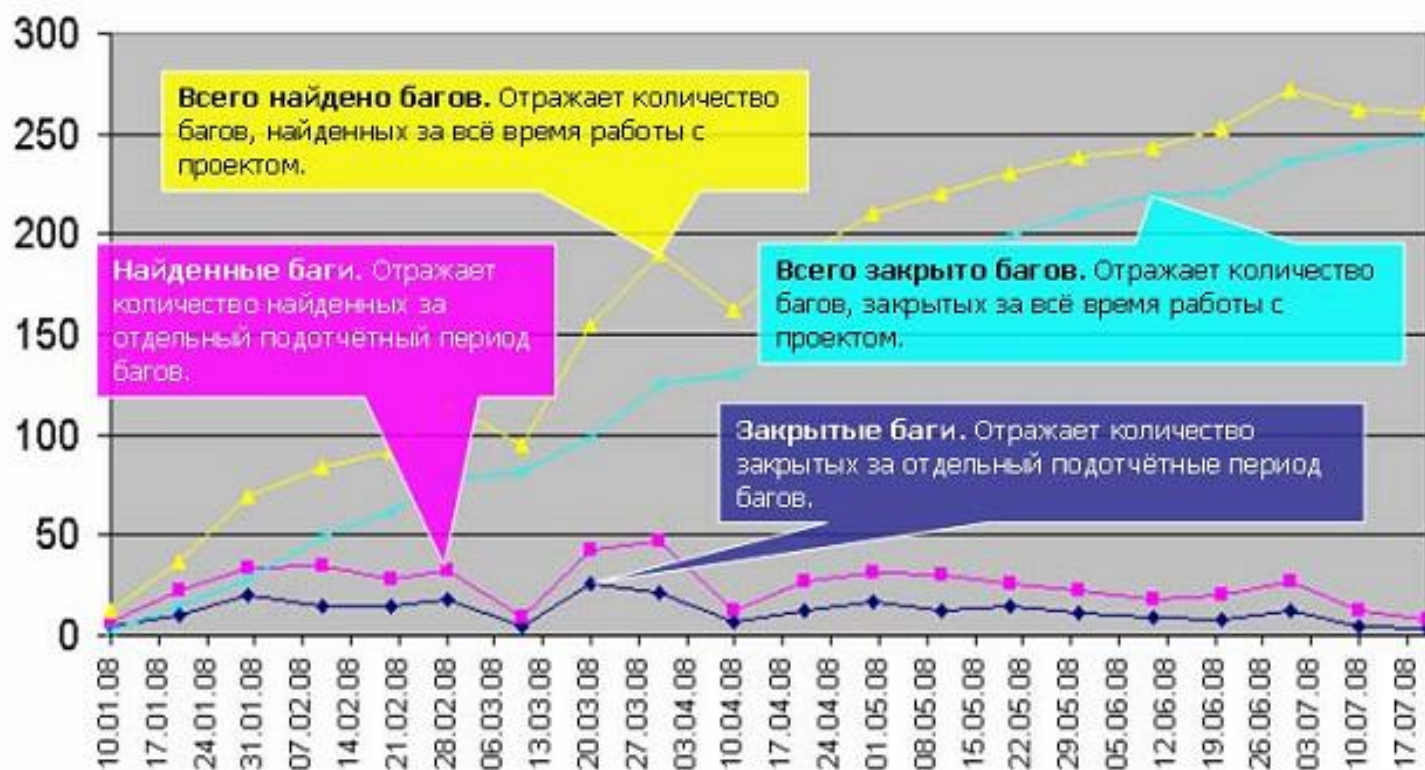
Статистика по всем ошибкам (*all bugs statistics*)

- Здесь приводится сводная таблица, содержащая информацию об ошибках, с которыми команде тестировщиков приходилось иметь дело *за всё время работы с проектом.*

Статус	Количество	Важность			
		Критическая	Высокая	Средняя	Минимальная
Исправлено	280	30	70	100	80
Проверено	170	50	20	50	50
Открыто заново	20	0	10	10	0
Найдено	360	50	110	80	120
Отклонено	40	0	0	10	30

Статистика по всем ошибкам (*all bugs statistics*)

Статистика по всем ошибкам также отражается в виде графика:



Кому необходим отчет об ошибках?

- Отчёт о результатах тестирования в основном нужен:
 - Менеджеру проекта
 - Лидеру команды разработчиков
 - Лидеру команды тестировщиков
 - Заказчику
- Обычно, TRR предоставляется для ознакомления всей проектной команде и заказчику

Менеджер проекта

- На основе краткого описания и статистики найденных багов он делает выводы о производительности работы проектной команды и текущем состоянии качества проекта.
- Расписание позволяет ему определить загруженность отдельных сотрудников в проектной команде.
- Список новых ошибок позволяет ему быстро оценить необходимость принятия мер по корректировке развития проекта.
- Рекомендации позволяют ему увидеть ещё одну точку зрения на происходящее в проекте.
- Т.о. информация из отчёта о результатах тестирования позволяет менеджеру проектов:
 - улучшить процесс управления, разработки и тестирования;
 - перераспределить ресурсы проекта;
 - сформировать собственный отчёт для руководства и заказчика.

Лидер команды разработчиков

- Несмотря на то, что разработчики не очень любят следовать рекомендациям тестировщиков, содержащаяся в отчёте о результатах тестирования информация помогает лидеру команды разработчиков пересмотреть некоторые вопросы повышения качества создаваемого кода и принять решение о возможном перераспределении обязанностей внутри его команды.

Лидер команды тестировщиков

- В процессе сбора информации и составления отчёта о результатах тестирования лидер команды тестировщиков полнее и глубже вникает в происходящее на проекте и внутри его команды, что позволяет ему увидеть текущую картину одновременно и более детально, и более широко.
- Такой взгляд позволяет обнаружить упущенные моменты и принять соответствующие решения, направленные на повышение качества проекта.

Заказчик

- Заказчика всегда интересует ответ на вопрос о том, куда идут его деньги. Подробный отчёт о результатах тестирования позволяет ему оценить, насколько его вложения оказались оправданными.
- В случае, если непосредственный заказчик выступает как чей-то подрядчик, отчёт о результатах тестирования позволяет ему отчитаться перед его заказчиком.
- В случае, если проект планируется принимать и внедрять по частям, заказчик может понять из отчёта, какие части проекта уже готовы к приёмке и внедрению.

Финальный отчёт о результатах тестирования

В конце работы с проектом формируется ещё один отчёт о результатах тестирования – финальный.

В дополнение к уже рассмотренным разделам такой отчёт включает описание и анализ существовавших на проекте проблем и найденных эффективных решений. Такой отчёт обсуждается на общем собрании проектной команды, где по результатам обсуждения формируются и документируются выводы, направленные на избежание в будущем проблем, возникших на данном проекте, а также направленные на накопление позитивного опыта с целью применения его в будущих или выполняемых параллельно проектах.

Контрольные вопросы:

1. Завершение тестирования. Что такое эвристика.
2. Описать все эвристики, определяющие завершение тестирования.
3. В чем заключается анализ результатов тестирования.
4. Что представляет собой TRR.
5. Структура TRR.
6. Кому необходим отчет о результатах тестирования.