

## Лабораторная работа №5

### Документирование результатов тестирования

*Цель:* составить итоговый отчет о результатах тестирования web-приложения.

*План занятия:*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

*Теоретические сведения*

Итоговый отчет о качестве проверенного функционала является неотъемлемой частью работы, которую каждый тестирующий должен выполнить по завершению тестирования.

Итоговый отчет можно разделить на части с соответствующей информацией:

1. Общая информация.
2. Сведения о том, кто и когда тестировал программный продукт.
3. Тестовое окружение.
4. Общая оценка качества приложения.
5. Обоснование выставленного качества.
6. Графическое представление результатов тестирования.
7. Детализированный анализ качества по модулям.
8. Список самых критичных дефектов.
9. Рекомендации.

Пример итогового отчета приведен на рисунке 6.1.

Далее рассмотрим подробно каждую часть итогового отчета.

Общая информация включает:

- название проекта;
- номер сборки;
- модули, которые подверглись тестированию (в случае, если тестировался не весь проект);
- виды тестов по глубине покрытия (Smoke Test, Minimal Acceptance Test, Acceptance Test), тестовые активности (New Feature Test, Regression Testing, Defect Validation);
- количество обнаруженных дефектов;
- вид рабочей тестовой документации (Acceptance Sheet, Test Survey, Test Cases).

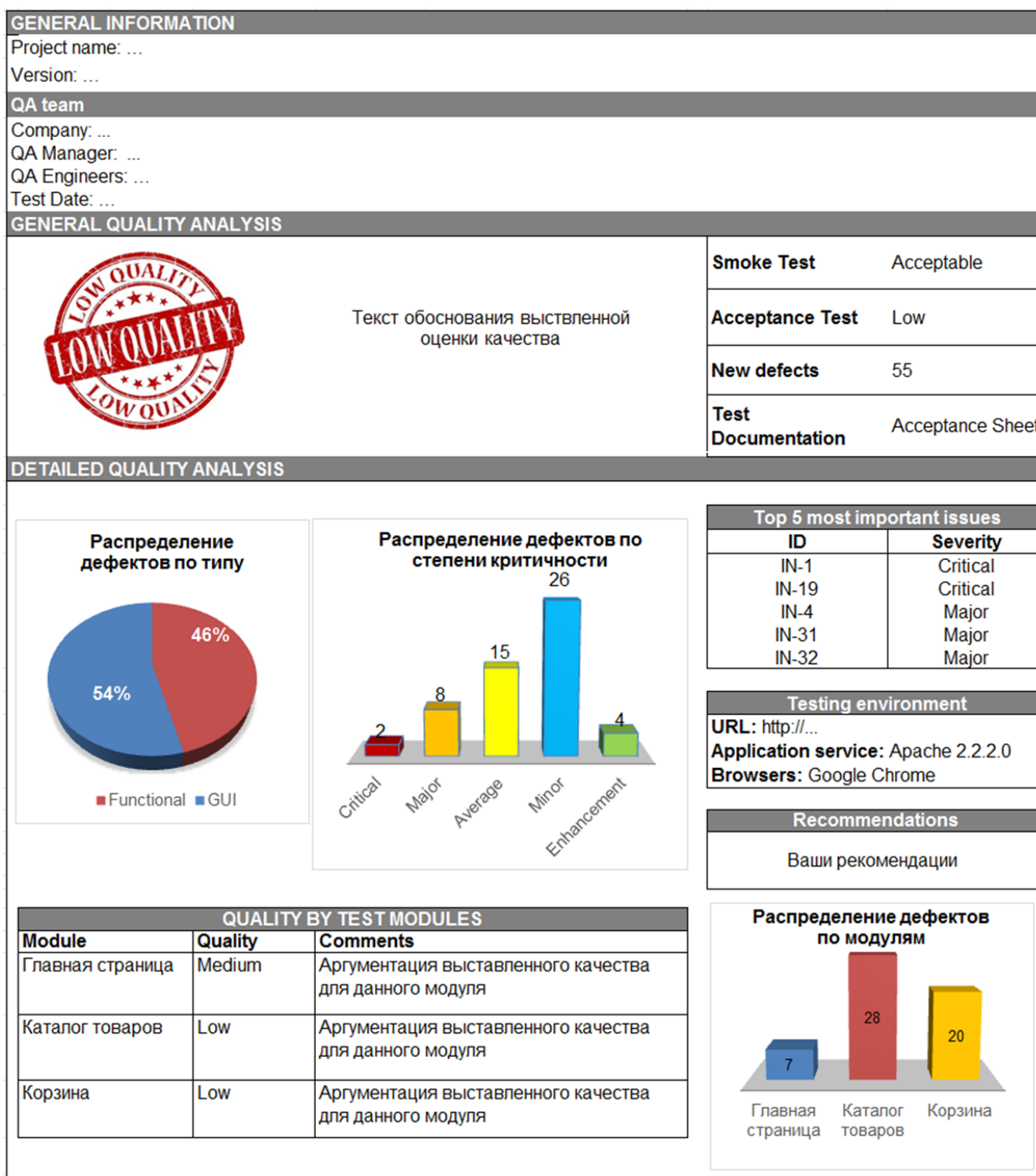


Рисунок 6.1 – Пример итогового отчета о результатах тестирования

Сведения о том, кто и когда тестировал программный продукт, включают информацию о команде тестирования с указанием контактных данных и временном интервале тестирования.

Тестовое окружение содержит ссылку на проект, браузер, операционную систему и другую информацию, конкретизирующую особенности конфигурации.

Общая оценка качества приложения выставляется на основании анализа результатов работы с приложением, количества внесенных дефектов, важности дефектов. Обязательно учитывается этап разработки проекта – то, что не критично в начале работы, становится важным при выпуске программного продукта. Уровни качества: высокое (High), среднее (Medium), низкое (Low).

Обоснование выставленного качества является наиболее важной частью отчета, т. к. здесь отражается общее состояние сборки, а именно:

- качество сборки на текущий момент;
- факторы, повлиявшие на выставление именно такого качества сборки: указание функционала, который заблокирован для проверки, перечисление наиболее критичных дефектов и объяснение их важности для пользователя или бизнеса заказчика;
- анализ качества проверенного функционала: улучшилось оно или ухудшилось по сравнению с предыдущей версией;
- если качество сборки ухудшилось, то обязательно должны быть указаны регрессионные места;
- наиболее нестабильные части функционала с указанием причин, по которым они таковыми являются.

Помимо вышеуказанных общих характеристик при выставлении и обосновании оценки качества программного продукта активно используются числовые характеристики качества – метрики. Пример обоснования с использованием метрик выглядит следующим образом: «Реализовано 79 % требований (в том числе 94 % важных), за последние три билда тестовое покрытие выросло с 63 до 71 %, а общий показатель прохождения тест-кейсов вырос с 85 до 89 %».

Метрики могут быть как прямыми (не требуют вычислений), так и расчетными (вычисляются по формуле). Типичные примеры прямых метрик – количество разработанных тест-кейсов, количество найденных дефектов и т. д.

Большинство общепринятых расчетных метрик могут быть собраны автоматически с использованием инструментальных средств управления проектами:

- процентное отношение (не)выполненных тест-кейсов ко всем имеющимся;
- процентный показатель успешного прохождения тест-кейсов;
- процентный показатель заблокированных тест-кейсов;
- плотность распределения дефектов;
- эффективность устранения дефектов;

- распределение дефектов по важности и срочности;
- метрики покрытия.

Простая расчетная метрика для определения показателя успешного прохождения тест-кейсов выглядит следующим образом:

$$T^{SP} = \frac{T^{Success}}{T^{Total}} \cdot 100 \%, \quad (6.1)$$

где  $T^{Success}$  – количество успешно выполненных тест-кейсов;

$T^{Total}$  – общее количество выполненных тест-кейсов.

Минимальные границы значений показателя успешного прохождения тест-кейсов на начальной фазе проекта равны 10 %, на основной фазе проекта – 40 %, на финальной фазе проекта – 85 %.

Метрику покрытия требований тест-кейсами вычисляют по формуле

$$R^{SimpleCoverage} = \frac{R^{Covered}}{R^{Total}} \cdot 100 \%, \quad (6.2)$$

где  $R^{Covered}$  – количество требований, покрытых хотя бы одним тест-кейсом;

$R^{Total}$  – общее количество требований.

Метрики являются мощнейшим средством сбора и анализа информации. Однако использование метрик требует ясного понимания их сущности, в противном случае может возникнуть ситуация использования «метрик ради метрик», когда многочисленные цифры и графики никто не может понять и правильно интерпретировать.

Графическое представление результатов тестирования способствует более полному и быстрому пониманию текстовой информации (см. рисунок 4.1).

Если необходимо продемонстрировать процентное соотношение, то целесообразно использовать круговые диаграммы (например, процентное соотношение функциональных дефектов и дефектов GUI).

Столбчатые диаграммы лучше подойдут там, где важно визуализировать количество дефектов в зависимости от степени их критичности или в зависимости от локализации (распределение дефектов по модулям).

Отразить в итоговом отчете динамику качества по всем сборкам лучше всего удастся с помощью линейного графика.

#### *Детализированный анализ качества по модулям*

В данной части отчета описывается более подробная информация о проверенных частях функционала, устанавливается качество каждой проверенной части функционала (модуля) в отдельности, дается аргументация выставленного уровня качества. Как правило, данный раздел отчета представляется в табличной форме. В зависимости от вида проводимых тестовых активностей эта часть отчета будет отличаться.

При оценке качества функционала на уровне Smoke теста оно может быть либо приемлемым (Acceptable), либо неприемлемым (Unacceptable). Если все наиболее

важные функции работают корректно, то качество всего функционала на уровне Smoke может быть оценено, как приемлемое.

Если это релизная или предрелизная сборка, то для выставления приемлемого качества на уровне Smoke не должно быть найдено функциональных дефектов.

В части о детализированной информации качества сборки следует более подробно описать проблемы, которые были найдены во время теста.

При оценке качества функционала на уровне Defect Validation указываются результаты валидации дефектов, а именно:

- общее количество всех дефектов, поступивших на проверку;
- количество неисправленных дефектов и их процент от общего количества;
- список дефектов, которые не были проверены и причины, по которым этого не было сделано;
- наглядная таблица с неисправленными дефектами.

По вышеуказанным результатам выставляется качество теста. Если процент неисправленных дефектов меньше 10 %, то качество приемлемое (Acceptable), если больше 10 %, то качество неприемлемое (Unacceptable).

При оценке качества функционала на уровне New Feature Test (полный тест нового функционала) качество отдельно проверенного функционала может быть высокое (High), среднее (Medium), низкое (Low).

Важно отдельно указывать информацию о качестве каждого модуля нового функционала с аргументацией выставленной оценки.

При оценке качества функционала на уровне Regression Testing нужно анализировать динамику изменения качества проверенной функциональности в сравнении с более ранними версиями сборки. Для этого приводится сравнительная характеристика каждой из частей функционала в сравнении с предыдущими версиями сборки, даются ясные пояснения о выставлении соответствующего качества каждой функции в отдельности. Так же как и у предыдущего вида тестов, качество этих может быть высокое (High), среднее (Medium), низкое (Low).

Список самых критичных дефектов содержит 3–5 ссылок на наиболее критичные дефекты с указанием их названия и уровня критичности.

Рекомендации включают краткую информацию о всех проблемах приложения с пояснениями, насколько оставшиеся проблемы являются критичными для конечного пользователя. Обязательно указывают функционал и дефекты, скорейшее исправление которых является наиболее приоритетным. Кроме того, если сборка является релизной или предрелизной, то любое ухудшение качества является критичным и важно это обозначить.

#### *Практическое задание:*

1. Составить итоговый отчет по результатам тестирования web-приложения.

2. Указать общую информацию о тестируемом продукте (название, номер сборки, виды выполненных тестов, количество обнаруженных дефектов, вид рабочей тестовой документации).
3. Указать, кто и когда тестировал программный продукт.
4. Описать тестовое окружение (ссылку на web-приложение, браузер).
5. Указать общую оценку качества протестированного приложения и подробно ее обосновать.
6. Графически (в виде круговой диаграммы) отразить процентное соотношение дефектов GUI и функциональных дефектов.
7. Графически (в виде столбчатой диаграммы) отразить распределение дефектов по различным степеням критичности.
8. Графически (в виде столбчатой диаграммы) отразить распределение дефектов по модулям.
9. Произвести детальный анализ качества всех модулей протестированного приложения с аргументацией выставленных уровней качества.
10. Привести список пяти наиболее критичных дефектов.
11. Сформулировать рекомендации по улучшению качества программного продукта.
12. Оформить отчет и защитить лабораторную работу.

*Содержание отчета:*

1. Цель работы.
2. Итоговый отчет о результатах тестирования web-приложения.
3. Выводы по работе.

*Контрольные вопросы:*

1. Какова структура итогового отчета о результатах тестирования?
2. Что содержится в разделе Общая информация?
3. Что содержится в разделе Тестовое окружение?
4. Как выставляется общая оценка качества приложения?
5. Как обосновать выставленную оценку качества?
6. Что такое метрика в тестировании?
7. Приведите примеры прямых метрик.
8. Приведите примеры расчетных метрик.
9. Для чего используется графическое представление результатов тестирования в итоговом отчете?
10. Что содержится в разделе Детализированный анализ качества?
11. Что содержится в разделе Рекомендации?