

Тест-дизайн и тест-аналитика
Урок 11

Тестовое покрытие





Оглавление

На этом уроке	3
Тестовое покрытие	3
Метрики покрытия кода	3
Метрика покрытия требований	4
Матрица трассировки	5
Варианты связей в матрице трассировки	6
Оценка покрытия с помощью матриц трассируемости	7
Создание и ведение матрицы	7
Сложности в работе с матрицей трассировки	8
Практическая польза от матрицы трассировки	8
Примеры матриц трассировки	9
Контрольные вопросы	10
Глоссарий	11
Задание	11
Дополнительные материалы	11
Используемые источники	11



На этом уроке

1. Познакомимся с тестовым покрытием и его особенностями.
2. Рассмотрим подробнее матрицу трассировки, ее примеры и пользу.

Тестовое покрытие

Тестовое покрытие — метрика оценки качества тестирования, отражающая плотность покрытия тестами требований и исполняемого кода.

Метрики покрытия кода

Метрика покрытия кода позволяет оценить, насколько полно строки кода, написанные разработчиком, покрыты модульными или интеграционными тестами. Вычисляется по формуле:

$$Tcov = (Ltc / Lcode) \cdot 100\%$$

Где:

- **Tcov** — тестовое покрытие;
- **Ltc** — количество строк кода, покрытых тестами;
- **Lcode** — общее количество строк кода.

Для разработчиков считается хорошей практикой писать и регулярно использовать модульные (unit) тесты для проверки корректности кода. Они помогают найти дефекты в модуле ещё до его передачи на тестирование. Существенно экономят время.

Чем больше модульных тестов написано, и чем больше строчек кода они покрывают, тем выше вероятность рано обнаружить дефект. Поэтому метрику покрытия кода нужно постоянно отслеживать. Делать это можно с помощью специальных инструментов. Один из самых популярных — SonarQube.

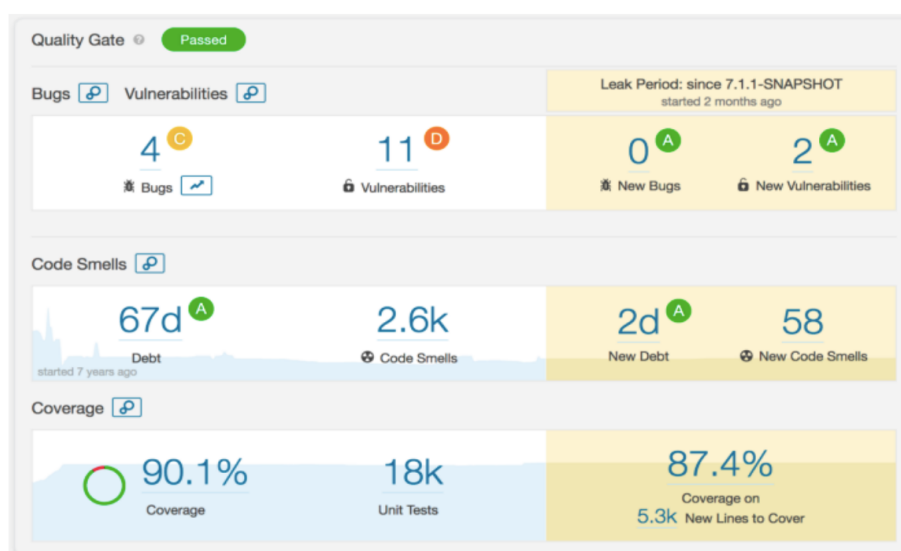
SonarQube — это платформа с открытым исходным кодом, предназначенная для непрерывного анализа и измерения качества кода и его покрытия. Её преимущества для разработчиков:

1. Поддержка языков Java, C, C++, C#, Objective-C, Swift, PHP, JavaScript, Python и других.



2. Предоставление отчётов о дублировании кода, соблюдении стандартов кодирования, покрытии кода модульными тестами, а также о возможных ошибках в коде, о плотности комментариев, техническом долге и так далее.
3. Сохранение истории метрик и построение графиков их изменения во времени.
4. Интеграция с IDE (Visual Studio, IntelliJ IDEA и Eclipse) с помощью плагина SonarLint.
5. Интеграция с внешними инструментами: JIRA, Mantis, LDAP, Fortify и другими.
6. Можно расширять функциональность с помощью сторонних плагинов.

Тестировщики могут использовать SonarQube, чтобы отслеживать качество кода и динамику изменений в нём.



На скриншоте количество дефектов и уязвимостей: в общем и в текущей итерации (выделено жёлтым). А также количество некачественных строчек кода и тестовое покрытие (в процентах и в строчках кода).

Таким образом, используя инструменты, можно достаточно быстро получить информацию о покрытии кода и внести изменения в процесс разработки.

Метрика покрытия требований

Если на проекте есть требования, их, как правило, много. Требования можно разбить (декомпозировать) на небольшие атомарные части, каждая из которых описывает определённую работу приложения. На каждую такую часть нужно написать тест-кейс, который проверит, что требование выполняется. Из-за



большого объёма может быть сложно отследить, для какого количества требований тест-кейсы уже написаны.

Чтобы определить качество и полноту покрытия кейсами, используют метрику покрытия требований. Она также полезна, если изменения вносят часто: можно отслеживать, что все они затронуты тест-кейсами. Вычисляется по формуле:

$$Tcov = (Lcov / Ltotal) \cdot 100\%$$

Где:

- **Tcov** — тестовое покрытие;
- **Lcov** — количество требований, проверяемых тест-кейсами;
- **Ltotal** — общее количество требований.

Чем выше процент метрики, тем качественнее будет проведено тестирование. Но бывает, что при высоком показателе значительная часть тест-кейсов проверяет одни и те же требования. Чтобы этого избежать и точно определить, какие требования покрыты тест-кейсами, а какие ещё нет, используют матрицу трассировки.

Матрица трассировки

Матрица трассировки (Traceability matrix) — это способ визуализации связей между элементами системы. Представляет собой таблицу, где:

- по одной грани указаны все требования на проекте, требования, которые должны быть протестированы в итерации или требования для определённого модуля либо функции,
- по другой грани — все написанные тест-кейсы.

На пересечении требования и тест-кейса ставится отметка о том, что требование покрывается тест-кейсом. Так удобно отслеживать, какие требования покрыты тест-кейсами, а какие нет, есть ли требования, для которых написано несколько тест-кейсов, исключать дублирующие или неэффективные тесты.

Матрицы трассировки используются не только для оценки покрытия, но и для определения связи между задачами на разработку, требованиями и тестовыми артефактами. Каждая строка матрицы содержит:



- номер и описание задачи на разработку;
- логический блок, к которому принадлежит задача (опционально);
- атомарное требование или приёмочный критерий;
- приоритет;
- номер и описание соответствующего тестового артефакта.

Такая трассируемость позволяет:

- визуализировать актуальное состояние реализации;
- разбивать требования на более атомарные и структурировать их;
- отслеживать, есть ли требования, на которые ещё не запланирована разработка (пропуск реализации);
- отслеживать, реализовано ли требование в данный момент;
- отслеживать, покрыто ли требование тест-кейсом (пропуск тестирования);
- наглядно отображать приоритезацию требований.

Варианты связей в матрице трассировки

Привязка требования и тест-кейса может быть:

- **1 к 1** — атомарное требование, которое покрывается одним тест-кейсом, тест-кейс покрывает только это требование;
- **1 к n** — требование, которое покрывается несколькими тест-кейсами, данные тест-кейсы покрывают только это требование;
- **n к n** — требование, которое покрывается несколькими тест-кейсами, данные тест-кейсы покрывают это и другие требования.

Если одно требование в матрице трассируемости покрывается несколькими тестами, можно говорить об избыточности тестирования. Надо проанализировать, насколько требование атомарно.

Оценка покрытия с помощью матриц трассируемости

Если для оценки покрытия служит метрика «отношение количества требований к количеству тестовых артефактов», то связи в матрице должны быть «1 к 1», а требования максимально декомпозированы.



«Пользователь должен иметь возможность изменять и форматировать письмо в текстовом редакторе».

Для покрытия такого требования одного тест-кейса недостаточно, но если в матрице будет прилинкован только один артефакт, визуально будет представление, что требование покрыто.

Решение:

- разделить требования на отдельные атомарные функции текстового редактора;
- для каждой функции написать приёмочный критерий;
- для каждого критерия создать тестовый артефакт;
- если несколько атомарных требований могут быть покрыты одним чек-листом, можно не делать избыточного дробления, сэкономя ресурсы.

У проектной документации может быть разный вид для каждой фичи: UML, схемы, диаграммы юз-кейсов и переходов. В этом случае можно составить отдельную матрицу для каждого модуля или фичи. Оценка покрытия также рассчитывается отдельно для каждого модуля или фичи.

Создание и ведение матрицы

1. Требования декомпозируются и приоритезируются. Результат этапа — структурированный и приоритезированный список всех требований по функциональности.
2. Постановка задач на разработку и внесение в матрицу соответствующих требований. Результат — отслеживаемость трассируемость требований и задач на разработку.
3. Разработка тест-кейсов и чек-листов. Матрица заполняется тест-кейсами.
4. Поддержка матрицы в актуальном состоянии. Изменения должны вноситься при любых модификациях требований. Нужно учитывать интеграционные связи между двумя матрицами, которые описывают разные фичи или модули. При изменении в одной обязательно проверять, не нужно ли исправить вторую.



Сложности в работе с матрицей трассировки

1. **Актуализация.** Матрица полезна, когда поддерживается её актуальность. Если не актуализировать матрицу, она будет вносить путаницу.
2. **Временные ресурсы.** На проекте может быть срочный релиз и работа с новыми требованиями в одно и то же время. Все ресурсы QA направляются на тестирование, а не на работу с требованиями. Из-за этого возрастает долг по тестовой документации.
3. **Эффективность.** Если проект небольшой и все требования оформлены в виде структурированного ТЗ, а тест-кейсы создаются на каждое требование сразу, матрица трассировки будет дублировать информацию. Это лишняя трата ресурсов.

Практическая польза от матрицы трассировки

- Удобно контролировать реализацию требований, отслеживать, что все они разработаны и протестированы, ничего не пропущено.
- Помогает команде QA отслеживать долг по тестовой документации.
- Можно контролировать изменённые требования.
- В процессах разработки и тестирования больше прозрачности.

Примеры матриц трассировки

Матрица требований к мобильному устройству с соотношением статусов:



Мобильное устройство: требование						
Файл Правка Вид Вставка Формат Данные Инструменты Дополнения Справка Все изменения на Диске сохранены						
fx Общие статусы						
	A	C	D	E	F	G
1	Требование	Тесты (ссылки)	Статус тестирования		Общие статусы в соотношении:	
2	Выбор режимов устройства осуществляется через боковое меню, которое можно вызвать свайпом справа на главном экране.	https://docs.google.com	Выполнено успешно		Выполнено успешно	40%
3	В режиме "Свободен", устройство использует максимальный уровень громкости, вибро-звонок и яркость цветового индикатора.	https://docs.google.com	Выявлены дефекты		Выявлены дефекты	30%
4	В режиме "Не беспокоить", устройство не использует звуковые уведомления, вибро-звонок отключен, цветовой индикатор загорается на все время активности устройства, связанной с уведомлениями.	https://docs.google.com	Выполнено успешно		Не тестировали	10%
6	В режиме "В офисе", устройство переводит уровень громкости равную 5 баллам, активность вибро-звонка соответствует уровню "2", световой индикатор ведет себя в соотв. со схемой "радуга", яркость экрана переводится в автоматический режим.	https://docs.google.com	Выполнено успешно			
7	В режиме "В самолёте", устройство отключает GSM, FM, WiFi, CMDA, BT, GPS модули, яркость экрана устанавливается равной 25%, уровень громкости 2, вибро-звонок отключен, индикатор отключен.	https://docs.google.com	Выполнено успешно			
8	В режиме "Энергосберегающий", устройство отключает FM, WiFi, CMDA, BT, GPS модули, яркость экрана устанавливается равной 25%, уровень громкости 4, вибро-звонок отключен, индикатор отключен.	https://docs.google.com	Выявлены дефекты			
9	Таймер: доступна возможность установки таймера на 1, 2, 3, 4, 5 часов после чего активируется выбранный режим устройства.	https://docs.google.com	Не тестировали			
10	В режиме "Конференция" устройство выключает звуковые и вибро-нотификации; уведомления об актуальных активностях для пользователя осуществляются путем отображения информации на	https://docs.google.com	Выявлены дефекты			

Матрица трассировки веб-приложения с разбиением по разделам:

Урок 11. Тестовое покрытие



Веб-приложение N									
Файл Правка Вид Вставка Формат Данные Инструменты Дополнения Справка Все изменения на Диске сохранены									
p. % .00 123 Arial 10 B I A [шаблоны] [список] [фильтры] [сумма] Py									
fx									
	A	B	C	D	E	F	G	H	I
3					Статусы			Ссылки	
4	№	Раздел	Функционал	Приоритет	Тех.Задание	Тест-анализ	Разр-ка	Тех.Задание	Тест-анализ
5	1	Пользователь							
6	1.1	Создание нового пользователя	Регистрация пользователя	Высокий	Неактуально	В процессе	Готово	https://docs.google.com	https://docs.google.com
7			Создание нового пользователя без активации						
8			Создание нового пользователя с активацией						
9	1.2	Профиль пользователя	Просмотр	Высокий	Готово	В процессе	Готово	https://docs.google.com	https://docs.google.com
10			Редактирование основных данных						
11			Редактирование подписок						
12			Редактирование интеграции с BTS						
13	1.3	Авторизация	Авторизация пользователя с разными статусами	Высокий	Готово	Готово	Готово	https://docs.google.com	https://docs.google.com
15	2.1	Чек-лист (как список тестов)							
16	2.1.1	Создание нового чек-листа (как список тестов)	Создание тестов вручную	Критичный	Готово	В процессе	Готово	https://docs.google.com	https://docs.google.com
17			Создание тестов через импорт						
18	2.1.2	Формирование данных о прохождении чек-листа (Запуск)	Запуск созданного чек-листа и его прохождение	Высокий	Готово	В процессе	Готово	https://docs.google.com	https://docs.google.com
19	2.1.2	Редактирование чек-листа	Внесение изменений в существующий чек-лист	Высокий	Неактуально	Неактуально	Неактуально	https://docs.google.com	https://docs.google.com
20			Удаление существующего чек-листа						

Матрица трассировки веб-продукта по модулям с итерациями:

Веб-продукт							
Файл Правка Вид Вставка Формат Данные Инструменты Дополнения Справка Все изменения на Диске сохранены							
p. % .00 123 Arial 10 B I A							
	A	B	C	D	E	F	G
1		Модули	Функционал	Приоритеты	Стратегия тестирования	Статус	Ссылка на ТЗ
2	Сервисы	Фото облако		Высокий	https://docs.google.com	Готов к релизу	https://docs.google.com
3		Аудио облако		Высокий	https://docs.google.com	Готов	https://docs.google.com
4		Видео облако		Высокий	https://docs.google.com	Готов	https://docs.google.com
5		Файлы и папки		Высокий	https://docs.google.com	Критичные ошибки	нет
6		Корзина		Низкий		Критичные ошибки	https://docs.google.com
7		Настройки	Основные	Низкий		Готов	
8			Личные данные	Низкий		Готов	
9			Безопасность	Высокий	https://docs.google.com	Критичные ошибки	
10			Рабочий стол	Низкий		Готов	
11			Управление местом	Низкий		Готов	https://docs.google.com
12	Главная страница/Трей	Публичные ссылки		Средний		Готов	https://docs.google.com
13		Бонусы		Высокий		Готов	https://docs.google.com
14		Активные процессы		Средний	https://docs.google.com	Готов	https://docs.google.com
15		Тех. поддержка		Низкий			
16		Часы		Низкий			
17		Управление		Средний			https://docs.google.com
18		Во весь экран		Низкий			
19		Рабочий стол		Средний	https://docs.google.com	Готов	Нет, и не будет

Матрица трассируемости интернет-магазина:



Интернет-магазин						
Файл Правка Вид Вставка Формат Данные Инструменты Дополнения Справка Все изменения на Диске сохранены						
<div> fx В С Д Е Ф Г </div>						
2	Раздел продукта	ТЗ, специфик	Шаблоны интерф	Стратегия тестиров	Ссылка на чеклист	Статус тестирования
3	Авторизация/регистрация					
4	Авторизация		http://drive.google.c		https://docs.google.co	Принято
5	Регистрация		http://drive.google.c		https://docs.google.co	Ошибки
6	Восстановление пароля	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Ошибки
7	Личный кабинет					
8	Аватар	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Ошибки
9	Прикрепление карты	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Принято
10	История заказов и их статусы	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Принято
11	Гости	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Принято
12	История адресов	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Принято
13	История заказов и зал славы					
14	История заказов	http://redmine.q	http://qlab.balsamiq	нет	нет	Не проведено
15	Зал славы	http://redmine.q	http://qlab.balsamiq	нет	нет	Не проведено
16	Начисление бонусов и подарков на всех этапах	http://redmine.q	http://qlab.balsamiq	нет	нет	Не проведено
17	Поиск товаров и навигация					
18	Поиск товаров	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Ошибки
19	Навигация по меню	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Ошибки
20	Выбор исключений по ингредиентам			https://docs.google.co	https://docs.google.co	Не принято
21	Выбор предпочтений по ингредиентам			https://docs.google.co	https://docs.google.co	Не принято
22	Апсейл с учетом данного выбора	http://drive.goog	http://drive.google.c	https://docs.google.co	https://docs.google.co	Не принято
23	Карточки блюд - информация	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Принято
24	Карточки блюд - работа с фото	https://docs.goc	http://drive.google.c	https://docs.google.co	https://docs.google.co	Принято
25	Корзина					
26	Добавление в корзину		http://drive.google.c	https://docs.google.co	https://docs.google.co	Принято

Контрольные вопросы

1. Что такое тестовое покрытие?
2. Для чего нам нужно тестовое покрытие?
3. Какие метрики используются для оценки тестового покрытия?
4. С помощью чего фиксируют и визуализируют тестовое покрытие?
5. Какой алгоритм создания матрицы трассировки?
6. Какие плюсы у матрицы трассировки?
7. В чем заключаются сложности работы с матрицей трассировки?

Глоссарий

Матрица трассировки — способ визуализации связей между элементами системы. Представляет собой таблицу, где по одной грани указаны требования, а по другой — написанные тест-кейсы.

Дополнительные материалы

1. [Матрица трассировки – в чем ценность для тестировщиков?](#)



2. [Пример матрицы трассировки требований](#)

Используемые источники

1. [Оценка тестового покрытия на проекте](#)
2. [Матрица трассабилити](#)