

- программные ресурсы (какое ПО необходимо команде тестировщиков, сколько копий и с какими лицензиями (если речь идёт о коммерческом ПО));
 - аппаратные ресурсы (какое аппаратное обеспечение, в каком количестве и к какому моменту необходимо команде тестировщиков);
 - человеческие ресурсы (сколько специалистов какого уровня и со знаниями в каких областях должно подключиться к команде тестировщиков в тот или иной момент времени);
 - временные ресурсы (сколько по времени займёт выполнение тех или иных работ);
 - финансовые ресурсы (в какую сумму обойдётся использование имеющихся или получение недостающих ресурсов, перечисленных в предыдущих пунктах этого списка); во многих компаниях финансовые ресурсы могут быть представлены отдельным документом, т. к. являются конфиденциальной информацией.
- **Расписание** (test schedule³³⁷). Фактически это календарь, в котором указано, что и к какому моменту должно быть сделано. Особое внимание уделяется т.н. «ключевым точкам» (milestones), к моменту наступления которых должен быть получен некий значимый ощутимый результат.
 - **Роли и ответственность** (roles and responsibility). Перечень необходимых ролей (например, «ведущий тестировщик», «эксперт по оптимизации производительности») и область ответственности специалистов, выполняющих эти роли.
 - **Оценка рисков** (risk evaluation). Перечень рисков, которые с высокой вероятностью могут возникнуть в процессе работы над проектом. По каждому риску даётся оценка представляемой им угрозы и приводятся варианты выхода из ситуации.
 - **Документация** (documentation). Перечень используемой тестовой документации с указанием, кто и когда должен её готовить и кому передавать.
 - **Метрики** (metrics³³⁸). Числовые характеристики показателей качества, способы их оценки, формулы и т. д. На этот раздел, как правило, формируется множество ссылок из других разделов тест-плана.

Метрики в тестировании являются настолько важными, что о них мы поговорим отдельно. Итак.



Метрика (metric³³⁸) — числовая характеристика показателя качества. Может включать описание способов оценки и анализа результата.

Сначала поясним важность метрик на тривиальном примере. Представьте, что заказчик интересуется текущим положением дел и просит вас кратко охарактеризовать ситуацию с тестированием на проекте. Общие слова в стиле «всё хорошо», «всё плохо», «нормально» и тому подобное его, конечно, не устроят, т.к. они предельно субъективны и могут быть крайне далеки от реальности. И совсем иначе выглядит ответ наподобие такого: «Реализовано 79 % требований (в т. ч. 94 % важных), за последние три спринта тестовое покрытие выросло с 63 % до 71 %, а общий показатель прохождения тест-кейсов вырос с 85 % до 89 %. Иными словами, мы полностью укладываемся в план по всем ключевым показателям, а по разработке даже идём с небольшим опережением».

Чтобы оперировать всеми этими числами (а они нужны не только для отчётности, но и для организации работы проектной команды), их нужно как-то вычислить. Именно это и позволяют сделать метрики. Затем вычисленные значения можно использовать для:

³³⁷ **Test schedule.** A list of activities, tasks or events of the test process, identifying their intended start and finish dates and/or times, and interdependencies. [ISTQB Glossary]

³³⁸ **Metric.** A measurement scale and the method used for measurement. [ISTQB Glossary]

- принятия решений о начале, приостановке, возобновлении или прекращении тестирования (см. выше раздел «Критерии» тест-плана);
- определения степени соответствия продукта заявленным критериям качества;
- определения степени отклонения фактического развития проекта от плана;
- выявления «узких мест», потенциальных рисков и иных проблем;
- оценки результативности принятых управленческих решений;
- подготовки объективной информативной отчётности;
- и т. д.

Метрики могут быть как прямыми (не требуют вычислений), так и расчётными (вычисляются по формуле). Типичные примеры прямых метрик — количество разработанных тест-кейсов, количество найденных дефектов и т. д. В расчётных метриках могут использоваться как совершенно тривиальные, так и довольно сложные формулы (см. таблицу 2.6.1).

Таблица 2.6.1 — Примеры расчётных метрик

Простая расчётная метрика	Сложная расчётная метрика
$T^{SP} = \frac{T^{Success}}{T^{Total}} \cdot 100\%, \text{ где}$ <p>T^{SP} — процентный показатель успешного прохождения тест-кейсов, $T^{Success}$ — количество успешно выполненных тест-кейсов, T^{Total} — общее количество выполненных тест-кейсов.</p> <p>Минимальные границы значений:</p> <ul style="list-style-type: none"> • Начальная фаза проекта: 10 %. • Основная фаза проекта: 40 %. • Финальная фаза проекта: 85 %. 	$T^{SC} = \sum_{Level}^{MaxLevel} \frac{(T_{Level} \cdot I)^{R_{Level}}}{B_{Level}}, \text{ где}$ <p>T^{SC} — интегральная метрика прохождения тест-кейсов во взаимосвязи с требованиями и дефектами, T_{Level} — степень важности тест-кейса, I — количество выполнений тест-кейса, R_{Level} — степень важности требования, проверяемого тест-кейсом, B_{Level} — количество дефектов, обнаруженных тест-кейсом.</p> <p>Способ анализа:</p> <ul style="list-style-type: none"> • Идеальным состоянием является непрерывный рост значения T^{SC}. • В случае отрицательной динамики уменьшение значения T^{SC} на 15 % и более за последние три спринта может трактоваться как недопустимое и являться достаточным поводом для приостановки тестирования.

В тестировании существует большое количество общепринятых метрик, многие из которых могут быть собраны автоматически с использованием инструментальных средств управления проектами. Например³³⁹:

- процентное отношение (не) выполненных тест-кейсов ко всем имеющимся;
- процентный показатель успешного прохождения тест-кейсов (см. «Простая расчётная метрика» в таблице 2.6.1);
- процентный показатель заблокированных тест-кейсов;
- плотность распределения дефектов;
- эффективность устранения дефектов;
- распределение дефектов по важности и срочности;
- и т. д.

³³⁹ «Important Software Test Metrics and Measurements — Explained with Examples and Graphs» [<http://www.softwaretestinghelp.com/software-test-metrics-and-measurements/>]

Как правило, при формировании отчётности нас будет интересовать не только текущее значение метрики, но и её динамика во времени, которую очень удобно изображать графически (что тоже могут выполнять автоматически многие средства управления проектами).

Некоторые метрики могут вычисляться на основе данных о расписании, например метрика «сдвига расписания»:

$$ScheduleSlippage = \frac{DaysToDeadline}{NeededDays} - 1, \text{ где}$$

ScheduleSlippage — значение сдвига расписания,

DaysToDeadline — количество дней до запланированного завершения работы,

NeededDays — количество дней, необходимое для завершения работы.

Значение *ScheduleSlippage* не должно становиться отрицательным.

Таким образом, мы видим, что метрики являются мощнейшим средством сбора и анализа информации. И вместе с тем здесь кроется опасность: ни при каких условиях нельзя допускать ситуации «метрик ради метрик», когда инструментальное средство собирает уйму данных, вычисляет множество чисел и строит десятки графиков, но... никто не понимает, как их трактовать. Обратите внимание, что к обоим метрикам в таблице 2.6.1 и к только что рассмотренной метрике *ScheduleSlippage* прилагается краткое руководство по их трактовке. И чем сложнее и уникальнее метрика, тем более подробное руководство необходимо для её эффективного применения.

И, наконец, стоит упомянуть про так называемые «метрики покрытия», т.к. они очень часто упоминаются в различной литературе.



Покрытие (coverage³⁴⁰) — процентное выражение степени, в которой исследуемый элемент (coverage item³⁴¹) затронут соответствующим набором тест-кейсов.

Самыми простыми представителями метрик покрытия можно считать:

- Метрику покрытия требований (требование считается «покрытым», если на него ссылается хотя бы один тест-кейс):

$$R^{SimpleCoverage} = \frac{R^{Covered}}{R^{Total}} \cdot 100\%, \text{ где}$$

R^{SimpleCoverage} — метрика покрытия требований,

R^{Covered} — количество требований, покрытых хотя бы одним тест-кейсом,

R^{Total} — общее количество требований.

- Метрику плотности покрытия требований (учитывается, сколько тест-кейсов ссылается на несколько требований):

$$R^{DensityCoverage} = \frac{\sum T_i}{T^{Total} \cdot R^{Total}} \cdot 100\%, \text{ где}$$

R^{DensityCoverage} — плотность покрытия требований,

T_i — количество тест-кейсов, покрывающих *i*-е требование,

T^{Total} — общее количество тест-кейсов,

R^{Total} — общее количество требований.

³⁴⁰ **Coverage, Test coverage.** The degree, expressed as a percentage, to which a specified coverage item has been exercised by a test suite. [ISTQB Glossary]

³⁴¹ **Coverage item.** An entity or property used as a basis for test coverage, e.g. equivalence partitions or code statements. [ISTQB Glossary]

- Метрику покрытия классов эквивалентности (анализируется, сколько классов эквивалентности затронуто тест-кейсами).

$$E^{Coverage} = \frac{E^{Covered}}{E^{Total}} \cdot 100\%, \text{ где}$$

$E^{Coverage}$ — метрика покрытия классов эквивалентности,

$E^{Covered}$ — количество классов эквивалентности, покрытых хотя бы одним тест-кейсом,

E^{Total} — общее количество классов эквивалентности.

- Метрику покрытия граничных условий (анализируется, сколько значений из группы граничных условий затронуто тест-кейсами).

$$B^{Coverage} = \frac{B^{Covered}}{B^{Total}} \cdot 100\%, \text{ где}$$

$B^{Coverage}$ — метрика покрытия граничных условий,

$B^{Covered}$ — количество граничных условий, покрытых хотя бы одним тест-кейсом,

B^{Total} — общее количество граничных условий.

- Метрики покрытия кода модульными тест-кейсами. Таких метрик очень много, но вся их суть сводится к выявлению некоей характеристики кода (количество строк, ветвей, путей, условий и т. д.) и определению, какой процент представителей этой характеристики покрыт тест-кейсами.



Метрик покрытия настолько много, что даже в ISTQB-гlossарии дано определение полутора десяткам таковых. Вы можете найти эти определения, выполнив в файле ISTQB-гlossария поиск по слову «coverage».

На этом мы завершаем теоретическое рассмотрение планирования и переходим к примеру — учебному тест-плану для нашего приложения «Конвертер файлов^{59}». Напомним, что приложение является предельно простым, потому и тест-план будет очень маленьким (однако, обратите внимание, сколь значительную его часть будет занимать раздел метрик).

ПРИМЕР ТЕСТ-ПЛАНА

Для того чтобы заполнить некоторые части тест-плана, нам придётся сделать допущения о составе проектной команды и времени, отведённом на разработку проекта. Поскольку данный тест-план находится внутри текста книги, у него нет таких типичных частей, как заглавная страница, содержание и т. п. Итак.

Цель

Корректное автоматическое преобразование содержимого документов к единой кодировке с производительностью, значительно превышающей производительность человека при выполнении аналогичной задачи.

Области, подвергаемые тестированию

(См. соответствующие разделы требований.)

- ПТ-1.*: дымовой тест.
- ПТ-2.*: дымовой тест, тест критического пути.
- ПТ-3.*: тест критического пути.
- БП-1.*: дымовой тест, тест критического пути.
- АК-1.*: дымовой тест, тест критического пути.
- О-4: дымовой тест.

- О-5: дымовой тест.
- ДС-*: дымовой тест, тест критического пути.

Области, не подвергаемые тестированию

- СХ-1: приложение разрабатывается как консольное.
- СХ-2, О-1, О-2: приложение разрабатывается на РНР указанной версии.
- АК-1: заявленная характеристика находится вблизи нижней границы производительности операций, характерных для разрабатываемого приложения.
- О-3: не требует реализации.
- О-6: не требует реализации.

Тестовая стратегия и подходы

Общий подход.

Специфика работы приложения состоит в однократном конфигурировании опытным специалистом и дальнейшем использовании конечными пользователями, для которых доступна только одна операция — размещение файла в каталоге-приёмнике. Потому вопросы удобства использования, безопасности и т.п. не исследуются в процессе тестирования.

Уровни функционального тестирования:

- Дымовой тест: автоматизированный с использованием командных файлов ОС Windows и Linux.
- Тест критического пути: выполняется вручную.
- Расширенный тест не производится, т.к. для данного приложения вероятность обнаружения дефектов на этом уровне пренебрежимо мала.

В силу кроссфункциональности команды значительного вклада в повышение качества можно ожидать от аудита кода, совмещённого с ручным тестированием по методу белого ящика. Автоматизация тестирования кода не будет применяться в силу крайне ограниченного времени.

Критерии

- Приёмочные критерии: успешное прохождение 100 % тест-кейсов уровня дымового тестирования и 90 % тест-кейсов уровня критического пути (см. метрику «Успешное прохождение тест-кейсов») при условии устранения 100 % дефектов критической и высокой важности (см. метрику «Общее устранение дефектов»). Итоговое покрытие требований тест-кейсами (см. метрику «Покрытие требований тест-кейсами») должно составлять не менее 80 %.
- Критерии начала тестирования: выход билда.
- Критерии приостановки тестирования: переход к тесту критического пути допустим только при успешном прохождении 100 % тест-кейсов дымового теста (см. метрику «Успешное прохождение тест-кейсов»); тестирование может быть приостановлено в случае, если при выполнении не менее 25 % запланированных тест-кейсов более 50 % из них завершились обнаружением дефекта (см. метрику «Стоп-фактор»).
- Критерии возобновления тестирования: исправление более 50 % обнаруженных на предыдущей итерации дефектов (см. метрику «Текущее устранение дефектов»).
- Критерии завершения тестирования: выполнение более 80 % запланированных на итерацию тест-кейсов (см. метрику «Выполнение тест-кейсов»).

Ресурсы

- Программные ресурсы: четыре виртуальных машины (две с ОС Windows 7 Ent x64, две с ОС Linux Ubuntu 14 LTS x64), две копии PHP Storm 8.
- Аппаратные ресурсы: две стандартных рабочих станции (8GB RAM, i7 3GHz).



- Человеческие ресурсы:
 - Старший разработчик с опытом тестирования (100%-я занятость на всём протяжении проекта). Роли на проекте: лидер команды, старший разработчик.
 - Тестировщик со знанием РНР (100%-я занятость на всём протяжении проекта). Роль на проекте: тестировщик.
- Временные ресурсы: одна рабочая неделя (40 часов).
- Финансовые ресурсы: согласно утверждённому бюджету. Дополнительные финансовые ресурсы не требуются.

Расписание

- 25.05 — формирование требований.
- 26.05 — разработка тест-кейсов и скриптов для автоматизированного тестирования.
- 27.05–28.05 — основная фаза тестирования (выполнение тест-кейсов, написание отчётов о дефектах).
- 29.05 — завершение тестирования и подведение итогов.

Роли и ответственность

- Старший разработчик: участие в формировании требований, участие в аудите кода.
- Тестировщик: формирование тестовой документации, реализация тестирования, участие в аудите кода.

Оценка рисков

- Персонал (вероятность низкая): в случае нетрудоспособности какого-либо из участников команды можно обратиться к представителям проекта «Каталогизатор» для предоставления временной замены (договорённость с менеджером «Каталогизатора» Джоном Смитом достигнута).
- Время (вероятность высокая): заказчиком обозначен крайний срок сдачи 01.06, потому время является критическим ресурсом. Рекомендуется приложить максимум усилий к тому, чтобы фактически завершить проект 28.05 с тем, чтобы один день (29.05) остался в запасе.
- Иные риски: иных специфических рисков не выявлено.

Документация

- Требования. Ответственный — тестировщик, дата готовности 25.05.
- Тест-кейсы и отчёты о дефектах. Ответственный — тестировщик, период создания 26.05–28.05.
- Отчёт о результатах тестирования. Ответственный — тестировщик, дата готовности 29.05.

Метрики

- Успешное прохождение тест-кейсов:

$$T^{SP} = \frac{T^{Success}}{T^{Total}} \cdot 100\%, \text{ где}$$

T^{SP} — процентный показатель успешного прохождения тест-кейсов,

$T^{Success}$ — количество успешно выполненных тест-кейсов,

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

Минимальные границы значений:

- Начальная фаза проекта: 10%.
- Основная фаза проекта: 40%.
- Финальная фаза проекта: 80%.

- Общее устранение дефектов:

$$D_{Level}^{FTP} = \frac{D_{Level}^{Closed}}{D_{Level}^{Found}} \cdot 100\%, \text{ где}$$

D_{Level}^{FTP} — процентный показатель устранения дефектов уровня важности $Level$ за время существования проекта

D_{Level}^{Closed} — количество устранённых за время существования проекта дефектов уровня важности $Level$,

D_{Level}^{Found} — количество обнаруженных за время существования проекта дефектов уровня важности $Level$.

Минимальные границы значений:

		Важность дефекта			
		Низкая	Средняя	Высокая	Критическая
Фаза проекта	Начальная	10%	40%	50%	80%
	Основная	15%	50%	75%	90%
	Финальная	20%	60%	100%	100%

- Текущее устранение дефектов:

$$D_{Level}^{FCP} = \frac{D_{Level}^{Closed}}{D_{Level}^{Found}} \cdot 100\%, \text{ где}$$

D_{Level}^{FCP} — процентный показатель устранения в текущем билде дефектов уровня важности $Level$, обнаруженных в предыдущем билде,

D_{Level}^{Closed} — количество устранённых в текущем билде дефектов уровня важности $Level$,

D_{Level}^{Found} — количество обнаруженных в предыдущем билде дефектов уровня важности $Level$.

Минимальные границы значений:

		Важность дефекта			
		Низкая	Средняя	Высокая	Критическая
Фаза проекта	Начальная	60%	60%	60%	60%
	Основная	65%	70%	85%	90%
	Финальная	70%	80%	95%	100%

- Стоп-фактор:

$$S = \begin{cases} \text{Yes}, & T^E \geq 25\% \ \&\& \ T^{SP} < 50\% \\ \text{No}, & T^E < 25\% \ || \ T^{SP} \geq 50\% \end{cases}, \text{ где}$$

S — решение о приостановке тестирования,

T^E — текущее значение метрики T^E ,

T^{SP} — текущее значение метрики T^{SP} .

- Выполнение тест-кейсов:

$$T^E = \frac{T^{Executed}}{T^{Planned}} \cdot 100\%, \text{ где}$$

T^E — процентный показатель выполнения тест-кейсов,

$T^{Executed}$ — количество выполненных тест-кейсов,

$T^{Planned}$ — количество тест-кейсов, запланированных к выполнению.

Уровни (границы):

- Минимальный уровень: 80 %.
- Желаемый уровень: 95–100 %.

- Покрытие требований тест-кейсами:

$$R^C = \frac{R^{Covered}}{R^{Total}} \cdot 100\%, \text{ где}$$

R^C — процентный показатель покрытия требования тест-кейсами,

$R^{Covered}$ — количество покрытых тест-кейсами требований,

R^{Total} — общее количество требований.

Минимальные границы значений:

- Начальная фаза проекта: 40 %.
- Основная фаза проекта: 60 %.
- Финальная фаза проекта: 80 % (рекомендуется 90 % и более).



Задание 2.6.а: поищите в Интернете более развёрнутые примеры тест-планов. Они периодически появляются, но и столь же быстро удаляются, т. к. настоящие (не учебные) тест-планы, как правило, являются конфиденциальной информацией.

На этом мы завершаем обсуждение планирования и переходим к отчётности, которая завершает цикл тестирования.

ОТЧЁТ О РЕЗУЛЬТАТАХ ТЕСТИРОВАНИЯ



Отчёт о результатах тестирования (test progress report³⁴², test summary report³⁴³) — документ, обобщающий результаты работ по тестированию и содержащий информацию, достаточную для соотнесения текущей ситуации с тест-планом и принятия необходимых управленческих решений.

К низкоуровневым задачам отчётности в тестировании относятся:

- оценка объёма и качества выполненных работ;
- сравнение текущего прогресса с тест-планом (в том числе с помощью анализа значений метрик);
- описание имеющихся сложностей и формирование рекомендаций по их устранению;
- предоставление лицам, заинтересованным в проекте, полной и объективной информации о текущем состоянии качества проекта, выраженной в конкретных фактах и числах.

Как и любой другой документ, отчёт о результатах тестирования может быть качественным или обладать недостатками. Качественный отчёт о результатах тестирования обладает многими свойствами качественных требований^[44], а также расширяет их набор следующими пунктами:

- Информативность (в идеале после прочтения отчёта не должно оставаться никаких открытых вопросов о том, что происходит с проектом в контексте качества).
- Точность и объективность (ни при каких условиях в отчёте не допускается искажение фактов, а личные мнения должны быть подкреплены твёрдыми обоснованиями).

³⁴² **Test progress report.** A document summarizing testing activities and results, produced at regular intervals, to report progress of testing activities against a baseline (such as the original test plan) and to communicate risks and alternatives requiring a decision to management. [ISTQB Glossary]

³⁴³ **Test summary report.** A document summarizing testing activities and results. It also contains an evaluation of the corresponding test items against exit criteria. [ISTQB Glossary]