Оглавление

[Верификация и валидация 2](#_Toc483142787)

[Верификация 3](#_Toc483142788)

[1.1.1. Степень покрытия программного кода тестами 3](#_Toc483142789)

[1.1.2. Возможные формы представления отчетов о прохождении тестов 4](#_Toc483142790)

[1.1.3. Возможные формы отчетов о покрытии 5](#_Toc483142791)

[1.1.4. Отчеты о проблемах 6](#_Toc483142792)

[1.1.5. Структура отчетов о проблемах, их трассировка на программный код и документацию 6](#_Toc483142793)

[1.2. Валидация 8](#_Toc483142794)

[1.2.1. Отчет о проведении валидации 8](#_Toc483142795)

[Формальные инспекции 9](#_Toc483142796)

[1.3. Задачи и цели проведения формальных инспекций 9](#_Toc483142797)

[1.3.1. Инициализация 10](#_Toc483142798)

[1.3.2. Планирование 11](#_Toc483142799)

[1.3.3. Подготовка 12](#_Toc483142800)

[1.3.4. Обсуждение 13](#_Toc483142801)

[1.3.5. Завершение 15](#_Toc483142802)

[1.4. Документирование процесса формальной инспекции 16](#_Toc483142803)

[1.4.1. Бланк инспекции 17](#_Toc483142804)

[1.4.2. Жизненный цикл инспектируемого документа 19](#_Toc483142805)

[1.5. Формальные инспекции программного кода 20](#_Toc483142806)

[1.5.1. Особенности этапа просмотра инспектируемого кода 20](#_Toc483142807)

[1.5.2. Особенности этапа проведения собрания 21](#_Toc483142808)

[1.5.3. Особенности этапа завершения и повторной инспекции 21](#_Toc483142809)

[1.6. Формальные инспекции проектной документации 22](#_Toc483142810)

[1.6.1. Особенности этапа просмотра документации 22](#_Toc483142811)

[1.6.2. Особенности этапа завершения 23](#_Toc483142812)

# Верификация и валидация

**Верификация и валидация** являются видами деятельности, направленными на контроль качества программного обеспечения и обнаружение ошибок в нем. Имея общую цель, они отличаются источниками проверяемых в их ходе свойств, правил и ограничений, нарушение которых считается ошибкой.

**Верификация** проверяет соответствие одних создаваемых в ходе разработки и сопровождения ПО артефактов другим, ранее созданным или используемым в качестве исходных данных, а также соответствие этих артефактов и процессов их разработки правилам и стандартам. В частности, верификация проверяет соответствие между нормами стандартов, описанием требований (техническим заданием) к ПО, проектными решениями, исходным кодом, пользовательской документацией и функционированием самого ПО. Кроме того, проверяется, что требования, проектные решения, документация и код оформлены в соответствии с нормами и стандартами, принятыми в данной стране, отрасли и организации при разработке ПО, а также - что при их создании выполнялись все указанные в стандартах операции, в нужной последовательности. Обнаруживаемые при верификации ошибки и дефекты являются расхождениями или противоречиями между несколькими из перечисленных документов, между документами и реальной работой программы, между нормами стандартов и реальным процессами разработки и сопровождения ПО. При этом принятие решения о том, какой именно документ подлежит исправлению (может быть, и оба) является отдельной задачей.

**Валидация** проверяет соответствие любых создаваемых или используемых в ходе разработки и сопровождения ПО артефактов нуждам и потребностям пользователей и заказчиков этого ПО, с учетом законов предметной области и ограничений контекста использования ПО. Эти нужды и потребности чаще всего не зафиксированы документально - при фиксации они превращаются в описание требований, один из артефактов процесса разработки ПО. Поэтому валидация является менее формализованной деятельностью, чем верификация. Она всегда проводится с участием представителей заказчиков, пользователей, бизнес-аналитиков или экспертов в предметной области - тех, чье мнение можно считать достаточно хорошим выражением реальных нужд и потребностей пользователей, заказчиков и других заинтересованных лиц. Методы ее выполнения часто используют специфические техники выявления знаний и действительных потребностей участников.

Верификация отвечает на вопрос "Делаем ли мы продукт правильно?", а валидация- на вопрос "Делаем ли мы правильный продукт?"

## Верификация



### Степень покрытия программного кода тестами

Степень покрытия программного кода тестами - важный количественный показатель, позволяющий оценить качество системы тестов, а в некоторых случаях - и качество тестируемой программной системы. Данные о степени покрытия помещаются в отчеты о покрытии, которые генерируются при выполнении тестов инструментальными средствами, поддерживающими процесс тестирования, т.е. по сути, генерируются средой тестирования. Формат отчетов о покрытии обычно единый внутри проекта или нескольких проектов и часто зависит от особенностей инструментальных средств тестирования.

В отчете о покрытии в стандартизированной форме указываются участки программного кода тестируемой системы (или ее части), которые не были выполнены во время выполнения тестовых примеров, т.е. не были покрыты тестами. Причины непокрытия анализируются тестировщиками, по результатам анализа составляются отчеты о проблемах и запросы на изменение - документы, где описываются объекты разработки, которые необходимо изменить, и причины этих изменений.

### Возможные формы представления отчетов о прохождении тестов

В стандарте IEEE 829 отчет о прохождении тестов разделен на три различных документа и описан в разделах 9 (Test log), 10 (Test incident report) и 11 (Test summary report). В эти разделы включены соответственно общий отчет о прохождении тестов, отчет о проблемах, выявленных в результате выполнения тестов и общую статистику прохождения тестов. В данном курсе отчет о прохождении тестов считается единым документом, разделенным на три части:

* общая (заголовочная информация);
* результаты выполнения тестовых примеров (положительные и отрицательные);
* итоговая информация о выполнении тестовых примеров (общая статистика по выполненным тестам).

Заголовочная часть отчета о прохождении тестов служит для идентификации отчета и протоколирования того, какая часть разрабатываемой системы подвергалась тестированию, какая ее версия, какая конфигурация тестового стенда использовалась для выполнения тестов.

В заголовочную часть отчета о выполнении тестов обычно включается следующая информация:

1. Название проекта или тестируемой системы
2. Общий идентификатор группы тестовых примеров, включенных в отчет
3. Идентификатор тестируемого модуля или группы модулей и номера их версий
4. Ссылку на разделы и версии тест-требований или функциональных требований, по которым написаны тесты, для которых сгенерирован отчет
5. Время начала выполнения теста и его продолжительность
6. Конфигурацию тестового стенда, на которой выполнялся тест
7. Имена и фамилии автора тестов и/или лица, выполнявшего тесты.

Следующая часть отчета о прохождении тестов должна содержать информацию о результате выполнения каждого тестового примера – завершился ли он успешно или в результате его выполнения были выявлены какие-либо несоответствия с ожидаемым результатом. В некоторых проектах эта часть отчета может быть представлена в одной из двух форм – полной или краткой. Полная форма содержит всю информацию о тестовом примере, краткая – только информацию об обнаруженных в результате выполнения тестового примера несоответствий ожидаемых и реальных выходных значений.

Обычно каждая запись о результате прохождения каждого тестового примера в полной форме содержит следующую информацию:

1. Идентификатор тестового примера
2. Краткое описание тестового примера
3. Перечисление всех входных значений тестового примера
4. Перечисление всех ожидаемых и реальных выходных значений тестового примера
5. Для каждой пары «ожидаемое-реальное выходное значение» - информацию о совпадении/несовпадении этих значений
6. Сообщение о том, пройден или не пройден тестовый пример.

В краткой форме каждая запись обычно содержит следующую информацию:

1. Идентификатор тестового примера
2. Перечисление не совпавших ожидаемых и реальных выходных значений тестового примера
3. Для каждой пары «ожидаемое-реальное выходное значение» - информацию о совпадении/несовпадении этих значений
4. Сообщение о том, пройден или не пройден тестовый пример.
5. Степень покрытия программного кода тестами – важный количественный показатель, позволяющий оценить качество системы тестов, а в некоторых случаях – и качество тестируемой программной системы. Данные о степени покрытия помещаются в отчеты о покрытии, генерируемые при выполнении тестов инструментальными средствами, поддерживающими процесс тестирования, т.е. по сути, генерируются средой тестирования (). Формат отчетов о покрытии обычно един внутри проекта или нескольких проектов и часто зависит от особенностей инструментальных средств тестирования.
6. В отчете о покрытии в стандартизированной форме указываются участки программного кода тестируемой системы (или ее части), которые не были выполнены во время выполнения тестовых примеров, т.е. не были покрыты тестами. Причины непокрытия анализируются тестировщиками, по результатам анализа составляются отчеты о проблемах и запросы на изменение – документы, в которых описывается объекты разработки, которые необходимо изменить и причины этих изменений.

### Возможные формы отчетов о покрытии

Типичный отчет о покрытии представляет собой список структурных элементов покрываемого программного кода (функций или методов), содержащий для каждого структурного элемента следующую информацию:

* Название функции или метода
* Тип покрытия (по строкам, по ветвям, MC/DC или иной)
* Количество покрываемых элементов в функции или методе (строк, ветвей, логических условий)
* Степень покрытия функции или метода (в процентах или в абсолютном выражении)
* Список непокрытых элементов (в виде участков непокрытого программного кода с номерами строк)

Кроме того, отчет о покрытии содержит заголовочную информацию, позволяющую идентифицировать отчет и общий итог – общую степень покрытия всех функций, для которых собирается информация о покрытии.

### Отчеты о проблемах

Каждое несоответствие с требованиями, найденное тестировщиком, должно быть задокументировано в виде отчета о проблеме. Вероятность обнаружения и исправления ошибки, вызвавшей это несоответствие, зависит от того, насколько качественно она задокументирована. Отчеты о проблемах могут поступать не только от тестировщиков, но и от специалистов технической поддержки или пользователей, однако их общая цель – указать на наличие проблемы в системе, которая должна быть устранена. Если отчет составлен некорректно, разработчик не сможет устранить проблему, поэтому можно считать этот отчет одним из самых важных документов в цепочке тестовой документации.

Главное, что должно быть включено в отчет об ошибке, это:

* Способ воспроизведения проблемы. Для того, чтобы разработчик смог устранить проблему, он должен разобраться в ее причинах, самостоятельно воспроизведя ее (и, возможно, не один раз). Один из самых тяжелых случаев в процессе разработки возникает при невоспроизводимой проблеме, т.е. проблеме, у которой точно неизвестен способ ее вызывать. Нахождение такого способа – одна из самых нетривиальных задач в работе тестировщика.
* Анализ проблемы с кратким ее описанием. Лучше всего приводить описание в тех же терминах, в которых составлены требования на часть системы, в которой обнаружена проблема. В этом случае минимизируется вероятность недопонимания сути проблемы.

Любой отчет о проблеме должен быть составлен немедленно после ее обнаружения. Если отчет будет составлен спустя значительное время, повышается вероятность того, что в него не попадет какая-либо важная информация, которая поможет устранить причину проблемы в кратчайшие сроки.

### Структура отчетов о проблемах, их трассировка на программный код и документацию

Структура отчёта о проблеме в целом мало различается в различных проектах, изменения обычно касаются только порядка и имен следования полей. Некоторые поля могут отражать специфику данного конкретного проекта, однако обычно эти поля следующие:

* Объект, в котором найдена проблема. Здесь помещается максимально полная информация – для документации это название документа, раздел, автор, версия. Для исходных текстов это имя модуля, имя функции/метода или номера строк, версия.
* Выпуск и версия системы. Определяет место, откуда был взят объект с обнаруженной проблемой. Обычно требуется отдельная идентификация версии системы (а не только версии исходных текстов), поскольку может возникнуть путаница с повторно выявленными проблемами. В этом случае, если проблема уже была когда-то выявлена разработчиком и потом вновь появилась из-за того, что в систему попала не самая последняя версия программного модуля, разработчик может решить, что ему пришел старый отчет и проблемы на самом деле не существует.
* Тип отчёта:
  + - Ошибка кодирования – код не соответствует требованиям.
    - Ошибка проектирования – тестировщик не согласен с проектной документацией.
    - Предложение – у тестировщика возникла идея, как можно усовершенствовать код.
    - Расхождение с документацией – поведение ПО не соответствует руководству пользователя или проектной документации, или вообще нигде не описано. При этом у тестировщика нет оснований объявлять, где именно находится ошибка.
    - Взаимодействие с аппаратурой – неверная диагностика плохого состояния устройства, ошибка в интерфейсе с устройством.
    - Вопрос – тестировщик не уверен, что это проблема и ему требуется дополнительная информация.
* Степень важности. Строгого критерия определения степени важности не существует и обычно это поле кодируют от 1 (незначительно) до 10 (фатально). Однако способов обоснованной оценки нет – очень сложно определить, насколько фатальной может оказаться, например, опечатка в один символ в руководстве пользователя.
* Суть проблемы. Краткое (не более 2 строчек) определение проблемы. Даже если две проблемы очень похожи, их описания должны различаться.
* Можно ли воспроизвести проблемную ситуацию? Ответ = Да, Нет, Не всегда. Последнее – если проблема носит нерегулярный характер. Нужно описывать, когда она проявляется, а когда – нет (например, не вовремя нажать не ту клавишу).
* Подробное описание проблемы и способ её воспроизведения. При этом нужны подробности – и в описании условий воспроизведения, и в описании причины объявления получившейся реакции ошибкой.

## [Валидация](http://gmpnews.ru/terminologiya/validation/)

[Валидация](http://gmpnews.ru/terminologiya/validation/) состоит из таких процессов:

— квалификация проектной документации ([Design Qualification](http://gmpnews.ru/kvalifikaciya-proekta/) — [DQ](http://gmpnews.ru/kvalifikaciya-proekta/)) — проверка описания и разработки системы;  
— квалификация инсталляции (монтажа) ([Installation Qualification](http://gmpnews.ru/kvalifikaciya-montazha/) — [IQ](http://gmpnews.ru/kvalifikaciya-montazha/)) — проверка способности инфраструктуры системы поддерживать работу системы;  
— [квалификация функционирования](http://gmpnews.ru/terminologiya/kvalifikaciya-funkcionirovaniya/) ([Operational Qualification](http://gmpnews.ru/terminologiya/kvalifikaciya-funkcionirovaniya/), [OQ](http://gmpnews.ru/terminologiya/kvalifikaciya-funkcionirovaniya/)) — проверка способности функционировать согласно требованиям;  
— [квалификация эксплуатации](http://gmpnews.ru/terminologiya/kvalifikaciya-ekspluatacii/) ([Performance Qualification](http://gmpnews.ru/terminologiya/kvalifikaciya-ekspluatacii/) — [PQ](http://gmpnews.ru/terminologiya/kvalifikaciya-ekspluatacii/)) — проверка способности компании использовать систему.

### Отчет о проведении валидации

Отчет о проведении валидации включает в себя следующие положения, информацию, документы:

1. Объект валидации и его идентификация, дата (период) и место проведения.

2. Цель и вид валидации.

3. Идентификация валидаторов (ФИО, должность, подпись, дата);

4. Исходная информация:

4.1. Общая характеристика объекта, включая критические параметры.

4.2. Перечень документации (регламенты, фармстатьи, проектная документации, инструкции, спецификации, сертификаты, паспорта и др.).

4.3. Перечень методик проведения испытаний (измерений, отбора проб и др.) и критериев оценки результатов.

4.4. Сведения о привлеченных организациях или экспертах.

5. Сведения о калибровке/поверке:

5.1. Средств измерений (приборы, датчики, весы и др.), установленных в оборудовании, инженерных системах, помещениях и др.

5.2. Средств измерений, используемых при проведении валидации/ квалификации.

6. Документы:

6.1. Валидационные протоколы всех стадий квалификации (DQ,IQ,OQ,PQ) и валидации процессов (PV), или ссылка на них с указанием места хранения.

6.2. Протоколы (отчеты и др.) с данными и результатами испытаний, отбора проб и т.п.

7. Анализ полученных результатов, в т.ч. по:

7.1. Проверке критических условий и параметров.

7.2. Выявленным отклонениям (изменениям), требующим действий по корректировке.

7.3. Условиям охраны труда и технике безопасности.

8. Вывод по результатам валидации.

9. Сроки проведения повторной плановой валидации.

# Формальные инспекции

## Задачи и цели проведения формальных инспекций

Не всегда возможна разработка автоматических или хотя бы четко формализованных ручных тестов для проверки функциональности программной системы. В некоторых случаях выполнение тестируемого программного кода невозможно в условиях, создаваемых тестовым окружением. Такая ситуация возможна во встроенных системах, если программный код предназначен для обработки исключительных ситуаций, создаваемых только на реальном оборудование.

В тех случаях, когда верифицируется не программный код, а проектная документация на систему, которую нельзя «выполнить» или создать для нее отдельные тестовые примеры также обычно прибегают к методу экспертных исследований программного кода или документации на корректность или непротиворечивость.

Такие экспертные исследования обычно называют инспекциями или просмотрами. Существует два типа инспекций – неформальные и формальные.

При неформальной инспекции автор некоторого документа или части программной системы передает его эксперту, а тот, ознакомившись с документом, передает автору список замечаний, которые тот исправляет. Сам факт проведения инспекции и замечания, как правило, нигде отдельно не сохраняются, состояние исправлений по замечаниям также нигде не отслеживается.

Формальная инспекция, напротив, является четко управляемым процессом, структура которого обычно четко определяется соответствующим стандартом проекта. Таким образом, все формальные инспекции имеют одинаковую структуру и одинаковые выходные документы, которые затем используются при разработке.

Факт начала формальной инспекции четко фиксируется в общей базе данных проекта. Также фиксируются документы, подвергаемые инспекции, списки замечаний, отслеживаются внесенные по замечаниям изменения. Этим формальная инспекция похожа на автоматизированное тестирование – списки замечаний имеют много общего с отчетами о выполнении тестовых примеров.

В ходе формальной инспекции группой специалистов осуществляется независимая проверка соответствия инспектируемых документов исходным документам. Независимость проверки обеспечивается тем, что она осуществляется инспекторами, не участвовавшими в разработке инспектируемого документа. Входами процесса формальной инспекции являются инспектируемые документы и исходные документы, а выходами – материалы инспекции, включающие список обнаруженных несоответствий и решение об изменении статуса инспектируемых документов.

Этапы формальной инспекции и роли ее участников

Как правило, процесс формальной инспекции состоит из пяти фаз: инициализация, планирование, подготовка (экспертиза), обсуждение, завершение. В некоторых случаях подготовку и обсуждение целесообразно рассматривать не как последовательные этапы, а как параллельные подпроцессы. В частности, такая ситуация может иметь место при использовании автоматизированной системы поддержки проведения формальных инспекций. Процедура формальной инспекции проекта должна точно описывать порядок проведения формальных инспекций в данном проекте.

После устранения обнаруженных в ходе формальной инспекции несоответствий процесс формальной инспекции повторяется, возможно, в другой форме и с другим составом участников. Процедура формальной инспекции должна регламентировать возможные формы проведения повторной инспекции в зависимости от объема и характера изменений, внесенных в объект инспекции. Как правило, допускается упрощение процесса повторной инспекции (проведение инспекции одним инспектором, отсутствие фазы обсуждения) при внесении в объект инспекции незначительных изменений относительно ранее инспектировавшейся версии.

### Инициализация

Руководитель проекта или его заместитель запрашивает из базы, хранящей все данные проекта (например, из системы конфигурационного управления, см. тему 11) список объектов, готовых к инспекции, выбирает объект инспекции, затем назначает участников формальной инспекции: автора, ведущего и одного или нескольких инспекторов. Ведущий также может выполнять роль инспектора; остальные участники выполняют только одну роль. На роль ведущего или инспектора не допускается назначать сотрудников, участвовавших в разработке объекта инспекции.

Обычно в роли автора выступает один из разработчиков объекта инспекции, но возможны ситуации, когда разработчик недоступен – например, переведен в другой проект или находится в отпуске. Тогда на роль автора назначается сотрудник, который будет исправлять обнаруженные несоответствия в инспектируемых документах. При инспектировании документов, разработанных заказчиком, автор может не назначаться.

Рекомендуется назначать не менее, чем двух инспекторов. Количество инспекторов может быть увеличено, если инспектируются документы, отличающиеся особой сложностью или новизной понятий, а также, если в качестве инспекторов привлекаются сотрудники с недостаточным опытом. Однако рекомендуемое общее число участников инспекции не должно превышать пяти.

В обоснованных случаях процедура формальной инспекции проекта может допускать проведение инспекции единственным инспектором, например, когда объект инспекции отличается особой простотой, и оцениваемые характеристики такого объекта инспекции тривиальны. Примером такого объекта инспекции может служить пакет результатов сбора структурного покрытия, получаемый после выполнения ранее проинспектированных тестов, для которого проверяется только состав пакета и согласованность версий.

В случае, если проводится повторная инспекция по сокращенной форме, ведущий самостоятельно инициирует процесс повторной инспекции без участия руководителя проекта. Процедура формальной инспекции проекта может разрешать ведущему самостоятельно инициировать процесс повторной инспекции (в том же составе участников), даже когда он проводится в полной форме, если это диктуется спецификой проекта.

### Планирование

Как только процесс формальной инспекции был инициирован, ведущий проверяет, что инспектируемые документы размещены в базе данных проекта, а их статус соответствует готовности к формальной инспекции. Если это не так, инспекция откладывается.

Затем он должен изменить статус инспектируемых документов так, чтобы отметить факт начала инспекции и ограничить доступ к инспектируемой документации. Во время инспекции изменения документов невозможно, а соответствующий статус сохраняется до конца инспекции. Далее будем называть этот статус Review.

После этого ведущий должен скопировать из базы данных проекта бланк инспекции и занести в него идентификаторы инспектируемых и исходных документов и номера их версий, список участников с указанием их ролей и дату фактического начала процесса инспекции, т.е. того момента, когда инспектируемые документы были переведены в состояние Review.

Ведущий должен оценить время, необходимое инспекторам для подготовки, и продолжительность обсуждения. Время, отводимое на этап подготовки, не может быть менее одного часа. Также ведущий должен определить дату, время и место обсуждения, если оно будет проходить в форме собрания. При этом может потребоваться согласование с другими участниками инспекции. Если оценка продолжительности обсуждения в форме собрания превышает 2 часа, то необходимо запланировать несколько собраний, каждое из которых будет длиться не более двух часов.

Процедура формальной инспекции проекта может допускать проведение повторной инспекции без собрания, если итогом предыдущей инспекции было решение о проведении повторной инспекции в сокращенной форме. Так же допускается не проводить собрание, если результаты формальной инспекции ведутся и хранятся в электронном виде. В этом случае, процедура формальной инспекции проекта должна регламентировать взаимодействия участников формальной инспекции между собой. Кроме того, процедура формальной инспекции проекта должна определять механизм подготовки, проведения обсуждения и принятия решения.

Подготовив бланк инспекции и определив время и место собрания, ведущий должен известить участников инспекции о времени и месте проведения собрания и разослать им подготовленный бланк инспекции.

Процедура формальной инспекции проекта может предусматривать использование бланка, заполненного в ходе предыдущей инспекции, если проводится повторная инспекция в сокращенной форме и при этом ведущий является единственным инспектором.

### Подготовка

Получив письмо или назначение с прикрепленным к нему бланком инспекции, инспекторы должны извлечь из базы данных проекта исходные и инспектируемые документы, используя указанные в бланке идентификаторы и номера версий. При этом инспекторы должны убедиться, что все документы находятся в соответствующем состоянии.

В ходе подготовки инспекторы детально изучают инспектируемые документы, руководствуясь списком контрольных вопросов. Обнаруженные несоответствия должны быть точно локализованы, сформулированы и записаны.

При проведении повторной инспекции в сокращенной форме допускается ограничиться анализом изменений по отношению к той версии объекта инспекции, которая инспектировалась на предыдущей инспекции. Если при этом обнаруживается, что имеются изменения, не связанные с зафиксированными замечаниями, то процесс инспекции прерывается и назначается новая инспекция в полной форме. Исключением из этого правила может быть случай, когда такие изменения заключаются в исправлении тривиальных ошибок, не затрагивающих сущности инспектируемых документов, таких, например, как опечатки в комментариях, не влияющие на смысл фразы.

Если при проведении повторной инспекции в сокращенной форме единственным инспектором, он считает, что объем изменений слишком велик, или изменения слишком сложны, он имеет право прервать процесс инспекции, известив руководителя проекта, с тем, чтобы была назначена новая инспекция в полной форме.

Автор, если он не является разработчиком объекта инспекции, должен в процессе подготовки детально с ним ознакомиться, чтобы быть готовым отвечать на вопросы инспекторов в ходе обсуждения, а после завершения инспекции устранить найденные несоответствия.

### Обсуждение

Обсуждение проводится в форме одного или нескольких собраний, каждое из которых продолжается не более двух часов. В один день рекомендуется проводить не более одного собрания. Если обсуждение не укладывается в запланированное число собраний, то назначаются дополнительные собрания. Для проведения собрания необходимо присутствие ведущего, хотя бы одного из инспекторов и, как правило, автора. Однако, ведущий может по своему усмотрению провести собрание в отсутствие автора, если тот болен или по какой-либо иной причине не может присутствовать на собрании, при условии, что ни один из инспекторов не обнаружил несоответствий, или их замечания очевидны и не требуют разъяснений со стороны автора, или с автором установлена телефонная связь. Если собрание было начато в отсутствие автора, а в дальнейшем возникла необходимость его присутствия, ведущий должен прервать и отложить собрание.

Собрание откладывается, если ни один из инспекторов не подготовился к обсуждению. Ведущий также может по своему усмотрению отложить собрание, если не подготовился или отсутствует хотя бы один из инспекторов.

В ходе обсуждения ведущий синхронизирует работу участников, зачитывая инспектируемый документ либо последовательно называя разделы или абзацы текста или элементы диаграмм, либо каким-то иным способом обеспечивает синхронный просмотр документа всеми участниками. По мере продвижения по документу инспекторы прерывают ведущего в тех местах, к которым у них имеются замечания. В случае отсутствия разногласий ведущий фиксирует несоответствие и продолжает продвижение по документу. При инспектировании документов небольшого объема ведущий, по своему усмотрению, может не синхронизировать просмотр документа всеми участниками, а просто опрашивать участников о наличии замечаний; такой опрос может быть совмещен с заполнением списка контрольных вопросов (см. ниже).

Если мнения участников по высказанному замечанию расходятся, то ведущий управляет дискуссией, последовательно предоставляя слово всем желающим высказаться, причем автор пользуется правом внеочередного предоставления слова. Если в результате дискуссии изменилась формулировка замечания, то ведущий записывает эту новую формулировку, затем зачитывает ее, и, если все участники с ней согласны, продолжает продвижение по документу.

Результатом дискуссии может также быть признание отсутствия проблемы. В этом случае ведущий убеждается в том, что все с этим согласны, и продолжает продвижение по документу.

Участники должны стремиться обозначить проблемы, но не искать их решения. Достижение консенсуса по спорным вопросам также не является целью дискуссии. Если имеется расхождение во мнениях, то должны быть зафиксированы все альтернативные мнения. Ведущий должен прервать дискуссию, если оценивает ее как непродуктивную.

Все участники обязаны уважительно относиться к оппонентам, не перебивать говорящего и высказываться тогда, когда ведущий предоставит им слово. Не допускаются параллельные обсуждения узким составом – каждый участник обязан адресовать свои высказывания всему собранию, а не соседу.

Необходимо также избегать критики и оценки квалификации коллег. Целью инспекции является повышение качества инспектируемых документов, а не оценка квалификации автора или других участников инспекции. Ведущий формальной инспекции не обладает преимуществом перед другими участниками обсуждения, он лишь организует этот процесс и фиксирует его результаты в бланке инспекции.

В ходе обсуждения необходимо в бланке инспекции проставить ответы на контрольные вопросы и зафиксировать замечания. Для этого ведущий последовательно зачитывает контрольные вопросы. При отсутствии у всех инспекторов замечаний, нарушающих сформулированное в вопросе свойство, против вопроса ставится отметка (галочка или крестик) в графе “Yes” или «Да»; в противном случае отметка ставится в графе “No” или «Нет», а в графе “Замечания” (или аналогичной) перечисляются номера соответствующих замечаний, записанных в таблице для замечаний, которая помещена в конце бланка инспекции.

Отметка в графе “N/A” (“Неприменимо”) ставится только в том случае, когда сформулированное в соответствующем вопросе свойство не может быть оценено для данного объекта инспекции; в этом случае в графе “Замечания” записывается обоснование невозможности оценить данное свойство.

Если при проведении повторной инспекции используется бланк от предыдущей инспекции, в котором уже проставлены ответы на контрольные вопросы, то эти ответы не исправляются, а в таблице для замечаний против зафиксированных ранее несоответствий делаются отметки об их устранении. В случае обнаружения новых несоответствий замечания записываются в таблицу после записанных ранее, а следующая повторная инспекция обязательно назначается в полной форме.

В конце обсуждения участники принимают решение о возможности принятия объекта инспекции в имеющейся версии, либо о необходимости внесения исправлений и проведения повторной инспекции в полной или сокращенной форме. Объект инспекции может быть принят в имеющейся версии только при отсутствии несоответствий. Решение о проведении повторной инспекции в сокращенной форме принимается только в том случае, если все участники с этим согласны. Если хотя бы один из участников настаивает на полной форме повторной инспекции, то повторная инспекция должна проводиться в полной форме. Мнение ведущего учитывается наравне с мнениями других участников. Принятое решение фиксируется ведущим на бланке инспекции и заверяется подписями всех участников, включая представителя службы качества, если он присутствовал на собрании.

Теоретически возможна ситуация, когда автор не согласен ни с одним из зафиксированных замечаний, а инспекторы настаивают, что несоответствия имеют место. В таком случае невозможно принять решение об изменении статуса инспектируемых документов, поэтому инспекция должна быть отложена, а решение проблемы вынесено за рамки процесса формальной инспекции.

На бланке инспекции также фиксируется продолжительность собрания и время, затраченное каждым из участников на подготовку.

Процедура формальной инспекции проекта может допускать отмену обсуждения и собрания, если ни у одного из инспекторов нет замечаний. Это возможно либо при проведении инспекции одним инспектором, одновременно являющимся ведущим, либо при использовании автоматизированной системы поддержки проведения формальных инспекций, которая обеспечит ведущего информацией о несоответствиях, обнаруженных каждым из инспекторов, и о том, что каждый из них подтвердил завершение изучения объекта инспекции. В этом случае заполнение бланка инспекции производится ведущим самостоятельно.

### Завершение

В конце собрания, по окончании обсуждения, инспекторы сдают ведущему свои рабочие материалы, которые включают в себя распечатки инспектируемых документов с пометками и бланки инспекции. Ведущий складывает эти материалы в прозрачную папку вместе с экземпляром бланка инспекции, заполненным в ходе обсуждения, причем титульный лист бланка инспекции должен лежать сверху, чтобы можно было по нему идентифицировать папки.

После собрания ведущий изменяет статус инспектируемых документов в базе данных проекта в соответствии с принятым решением – либо им присваивается статус «Принят», либо «Переработать».

В последнем случае необходима повторная инспекция, вид которой уточняется кратким комментарием.

## Документирование процесса формальной инспекции

Обычно, если предприятие ведет несколько проектов по разработке программных систем, процедура формальной инспекции регламентируется в виде стандарта предприятия. Это позволяет сотрудникам, участвующим в формальных инспекциях, легко адаптироваться при переходе из проекта в проект.

Однако каждый проект может иметь свою специфику. В силу этого рекомендуется разрабатывать для каждого проекта свою процедуру формальной инспекции. Процедура формальной инспекции проекта должна уточнять и дополнять настоящий стандарт с учетом специфики данного проекта. Процедура формальной инспекции проекта не должна противоречить требованиям настоящего стандарта.

Процедура формальной инспекции проекта должна точно описывать порядок проведения формальных инспекций в данном проекте.

В процедуре формальной инспекции проекта не рекомендуется дублировать общие положения настоящего стандарта, за исключением отдельных, особо важных моментов, таких, например, как изменение статуса инспектируемых документов.

В процедуре формальной инспекции проекта должны быть названы все идентификаторы состояний инспектируемых документов в базе данных проекта, с которыми приходится иметь дело участникам формальной инспекции. Как минимум, должны быть названы идентификаторы состояний, обозначающих:

* готовность документа к проведению инспекции;
* прохождение фаз планирования, подготовки и обсуждения;
* необходимость переработки документа;
* подтверждение соответствия исходным документам.

Процедура формальной инспекции проекта должна регламентировать минимальное количество инспекторов для каждого типа объектов инспекции, если в проекте инспектируются документы различного уровня сложности, требующие участия различного количества инспекторов.

Процедура формальной инспекции проекта должна регламентировать возможные формы проведения повторной инспекции в зависимости от объема и характера изменений, внесенных в объект инспекции. Как правило, допускается упрощение процесса повторной инспекции (выполнение инспекции одним инспектором, отсутствие фазы обсуждения) при внесении в объект инспекции незначительных изменений. Процедура формальной инспекции проекта может предусматривать использование бланка от предыдущей инспекции, если проводится повторная инспекция в сокращенной форме. Процедура формальной инспекции проекта может разрешать ведущему самостоятельно инициировать процесс повторной инспекции (в том же составе участников), даже когда он проводится в полной форме, если это диктуется спецификой проекта.

### Бланк инспекции

Бланк инспекции – основной документ, который заполняется в ходе проведения инспекций. Обычно он разрабатывается в ходе разработки стандартов проекта. Для каждого типа объектов инспекции в проекте должен быть разработан свой бланк инспекции.

Бланк инспекции состоит из трех основных частей:

* титульный лист;
* список контрольных вопросов;
* список несоответствий.

Кроме того, рекомендуется на всех страницах бланка, кроме первой, помещать колонтитул, включающий в себя как минимум номер бланка инспекции.

#### Титульный лист

Титульный лист предназначен для идентификации формальной инспекции и записи решения и обычно включает, как минимум, следующие элементы:

* слова «формальная инспекция»;
* идентификатор проекта;
* идентификатор типа объекта инспекции, например, «Тест», «Стандарт проекта» и т.п.;
* идентификатор версии бланка инспекции;
* идентификатор конфигурационной базы данных;
* место для записи идентификаторов каждого из инспектируемых документов;
* место для записи идентификаторов версий каждого из инспектируемых документов;
* место для записи идентификаторов каждого из исходных документов;
* место для записи идентификаторов версий каждого из исходных документов;
* место для записи даты начала инспекции;
* место для записи фактических даты и времени начала собрания;
* место (таблица) для записи фамилий участников инспекции с указанием их ролей и местами для подписи и записи времени, затраченного на подготовку;
* место для записи продолжительности собрания;
* место для фиксации принятого решения.

Идентификатор документа состоит из имени файла в базе данных проекта и полного пути к нему. Общие для разных документов элементы идентификации, такие, как путь или имя базы, могут быть вынесены в отдельные поля бланка.

Если процедурой формальной инспекции проекта предусмотрена возможность проведения повторной инспекции с использованием бланка от предыдущей инспекции, то титульный лист также должен включать следующие поля:

* место для записи даты проведения повторной инспекции;
* место для записи идентификаторов версий каждого из повторно инспектируемых документов;
* место для записи фамилии ведущего повторной инспекции;
* место для записи времени, затраченного ведущим на проведение повторной инспекции;
* место для фиксации принятого решения;
* место для подписи ведущего.

Все перечисленные элементы должны располагаться на одной странице.

#### Список контрольных вопросов

Список контрольных вопросов должен быть оформлен в виде таблицы, состоящей из следующих колонок:

* порядковый номер;
* текст вопроса;
* место для положительного ответа («Yes» или «Да»);
* место для отрицательного ответа («No» или «Нет»);
* место для ответа «N/A» или «Неприменимо»;
* место для ссылки на несоответствие.

Контрольные вопросы должны быть сформулированы таким образом, чтобы положительный ответ означал отсутствие несоответствий. Формулировки должны быть понятными, четкими и однозначными.

#### Список несоответствий

Список несоответствий должен быть оформлен в виде незаполненной таблицы с тремя колонками:

* для порядкового номера;
* для описания несоответствия;
* для отметки об исправлении.

#### Колонтитул

Колонтитул должен включать:

* идентификатор проекта;
* идентификатор версии бланка инспекции;
* место для записи идентификаторов хотя бы одного из инспектируемых документов;
* место для записи идентификаторов версий хотя бы одного из инспектируемых документов.

### Жизненный цикл инспектируемого документа

В процессе формальной инспекции существует 2 типа документов:

* документы проекта (целевой документ, исходный документ, поддерживающий документ);
* вспомогательные документы (отчет о проведенной инспекции, список контрольных вопросов, список обнаруженных проблем).

Вспомогательные документы возникают в процессе инспекции и могут изменяться в течение процесса инспекции. Титульный лист создается на стадии инициализации. Список обнаруженных проблем создается на стадии подготовки. Список контрольных вопросов заполняется на стадии обсуждения. После завершения процесса инспекции вспомогательные документы помещаются в архив и более не подлежат изменению. Вспомогательные документы хранятся в соответствии с установленными для них сроками.

В процессе формальной инспекции инспектируемый документ последовательно сменяет несколько состояний. В процессе разработки (до начала формальной инспекции) документ имеет состояние Active (Активный). В этом состоянии автор может обращаться к документу как по чтению, так и по записи. После того, как автор посчитал, что закончил работу над документом, он переводит документ в состояние Ready (Готов). Это означает, что документ готов к формальной инспекции. В состоянии Ready автор уже не может изменять документ. Следующим состоянием документа является Review (формальная инспекция). В это состояние документ помещается на стадии инициализации формальной инспекции. Перевод документа в состояние Review осуществляет ведущий. В состоянии Review доступ к документу возможен только для чтения для всех участников формальной инспекции. Если документ прошел формальную инспекцию (не было обнаружено проблем), то он переходит в состояние Approved (Утвержден). Перевод документа в состояние Approved осуществляет ведущий. В этом состоянии документ доступен только для чтения для всех участников формальной инспекции, а также для остальных участников проекта. Если же после формальной инспекции в целевом документе требуются исправления, документ переводится в состояние Update (Переработка). В этом состоянии автор имеет доступ к документу как по чтению, так и по записи. После переработки документа автор присваивает документу состояние Ready и процесс перехода по состояниям повторяется до тех пор, пока документ не будет переведен в состояние Approved. Если в инспектируемый документ не требуется вносить значительных изменений то, после того как ведущий убедится в том, что необходимые исправления были сделаны, целевой документ может быть переведен в состояние Approved.

Инспектируемый документ подлежит исправлению после завершения процесса инспекции. После исправления целевой документ может пройти повторную инспекцию. Таким образом, целевой документ может пройти несколько последовательных инспекций (совершить несколько витков жизненного цикла документов в процессе формальной инспекции). Общий жизненный цикл инспектируемого документа может изображен на .



Рис. 22 Жизненный цикл инспектируемого документа в процессе формальной инспекции

## Формальные инспекции программного кода

Процесс формальной инспекции программного кода подчиняется всем правилам, определенным для абстрактной формальной инспекции, однако, имеет некоторые особенности, связанные, в первую очередь со структурой инспектируемого программного кода, а также с тем, что обычно инспектируются участки кода, которые невозможно проверить при помощи автоматизированного тестирования, основанного на тестовых примерах.

### Особенности этапа просмотра инспектируемого кода

**Выделение ключевых точек и построение или использование таблиц трассировки.** Перед началом просмотра исходного кода рекомендуется отметить пункты требований, на соответствие которым проверяется исходный код, а также записать обоснования того, почему эти требования не могут быть проверены в автоматическом режиме. После этого можно переходить к просмотру собственно исходного кода. Все пометки, которые придется вносить в ходе инспектирования в исходный код необходимо делать не в файле, хранящемся в базе данных проекта, а в его копии, которая потом будет подшита к материалам инспекции. Копия может быть в том же формате, что и исходный файл, либо распечатана на бумаге или выведена в формат DOC, PDF или аналогичный, допускающий комментирование.

При помощи трассировочных таблиц в исходном коде определяются инспектируемые функции или методы, соответствующие необходимым требованиям. Участки кода выделяются и отмечаются меткой или номером соответствующего требования. Если участок кода соответствует требованиям, то необходимо отметить этот факт либо цветом выделения, либо соответствующим текстовым примечанием. Если участок кода имеет проблемы, этот факт должен быть отражен либо цветом выделения, либо ссылкой на соответствующий пункт списка замечаний в бланке инспекции.

В случае отсутствия трассировочных таблиц требований на исходный код рекомендуется делать пометки, поясняющие, почему именно данный участок кода реализует указанные требования. Такие пометки помогут на этапе обсуждения документа.

**Проверка стиля кодирования.** Отдельным объектом проверки при формальной инспекции программного кода является стиль кодирования. В большинстве проектов существуют стандарты, описывающие правила оформления исходных текстов программ и файлов данных. Неверный стиль кодирования не влияет на работоспособность программы в целом, но значительно затрудняет сопровождение и поддержку изменений в ходе дальнейшего развития системы. Поэтому отклонения от стиля кодирования в инспектируемых участках кода также должны отмечаться в тексте и в списке замечаний. В некоторых случаях проводят инспекции, целиком направленные на проверку стиля кодирования.

**Проверка надежности кода.** В некоторых случаях рекомендуется проверять наличие участков, гарантирующих робастность, даже если требования прямо не определяют необходимости обработки недопустимых значений. В случае, если потенциально возможна некорректная работа программы из-за отсутствия обработчиков неверных значений, рекомендуется отметить это в списке замечаний.

### Особенности этапа проведения собрания

**Распределение ролей.** В составе инспекторов желательно иметь хотя бы одного специалиста, представляющего себе особенности выполнения инспектируемого кода в реальной установке системы. Это особенно важно при тестировании встроенных систем, тестирование которых проводится на эмуляторах. Во время собрания такой специалист может помогать ведущему определять последовательность рассмотрения замечаний в случае их большого количества.

**Управление собранием.** При проведении собрания нецелесообразно зачитывать текст инспектируемого файла, как это обычно рекомендуется. Вместо этого ведущему лучше ограничиться перечислением имен функций или методов, либо, в случае, если в ходе инспекции проверяется соответствие исходного кода требованиям – перечислением номеров или идентификаторов требований. Инспектора при наличии замечаний по функции или требованию поднимают руку и зачитывают замечания.

### Особенности этапа завершения и повторной инспекции

**Документирование собрания.** Для облегчения труда автора инспектируемого документа по исправлению замечаний, каждое замечание, признанное на собрании существенным, рекомендуется точно трассировать на строки исходного кода и требований.

**Контроль за внесением изменений.** При повторной инспекции исходных текстов рекомендуется использовать специализированные инструментальные средства для сравнения файлов. Изменения по итогам инспекции должны вноситься только в те участки, к которым были высказаны замечания. В случае наличия других изменений ведущий вправе назначить новую инспекцию в полной форме.

## Формальные инспекции проектной документации

Процесс формальной инспекции проектной документации подчиняется всем правилам, определенным для абстрактной формальной инспекции, однако, имеет некоторые особенности, связанные, в первую очередь, с тем, что у проектной документации проверяется ее непротиворечивость и полнота. Точное определение этих терминов в рассматриваемом контексте затруднительно, поэтому под непротиворечивостью будем понимать отсутствие в проектной документации требований, с противоположным смыслом, согласно которым возможно несколько совершенно различных вариантов реализации программной системы. Под полнотой будем понимать достаточность требований для однозначной реализации поведения системы.

### Особенности этапа просмотра документации

При инспектировании требований к системе, как правило, рассматривается как «внешняя», так и «внутренняя» информация, касающаяся данного документа. Под внешней информацией понимается в первую очередь суть технических решений, принятых при разработке системы, те принципы, которые отличают ее от других систем. При этом проверяется согласованность требований с этими принципами. Под внутренней информацией понимается прежде всего внутренняя целостность и непротиворечивость документа – свойства, которые позволяют разрабатывать программный код, лишенный двусмысленностей и нестабильных участков. Главный вопрос, на который должен ответить инспектор при проверке внутренних свойств документа: "Определены ли требования так, чтобы коллектив разработчиков смог работать с ним?" или иначе "Эти требования недвусмысленны, полны и авторитетно выражены?". Процесс инспекции может помочь ответить на эти вопросы, а список контрольных вопросов, представленный ниже, может быть использован при проведении инспекций:

1. Является ли каждое требование совершенно недвусмысленным? (Если требование прочесть подряд несколько раз, делая ударение сначала на первом слове, потом - на втором, затем - на третьем и т.д., будет ли при этом меняться смысл этого тре­бо­ва­ния?)
2. Существует ли для каждого из установленных требований некоторый компетентный специалист, который сможет сказать после завершения разработки, выполнено ли данное тре­бование или нет? Определен ли метод решения этой проблемы в документации по требованиям?
3. Существует ли какие-либо не установленные или отсутствующие требования?
4. Существуют ли среди заданных такие требования, которые не являются не­об­хо­ди­мы­ми?
5. Если существуют какие-либо конфликтующие требования, известно ли понятное ре­ша­ющее правило, в каких ситуациях какому требованию следует отдавать пре­д­по­ч­те­ние?

### Особенности этапа завершения

**Влияние несогласованности документации на процесс разработки. Трассировка изменений на программный код.** Первичная инспекция проектной документации, как правило, проводится тогда, когда сама программная система еще не написана. Однако при проведении инспекции изменений в требованиях к уже работающей системе (например, при обновлении ее версии), может потребоваться комплексная одновременная инспекция документации и созданного на ее основе программного кода. При этом может возникнуть ситуация, когда изменения, которые требуется внести в документацию по результатам инспекции, требуют соответствующего изменения в программном коде. Решение этой проблемы достигается использованием трассировочных таблиц.

Несколько иная ситуация возникает в случае, когда комплексная инспекция проводится не после изменения требований, а после завершения всей цепочки изменений – после изменения функциональных требований, архитектуры, низкоуровневых требований и программного кода. В этом случае при обнаружении противоречивых требований необходимо выявить все части программной системы, которые реализуют эти требования. В случае, если разработка этих частей выполнялась разными людьми, могла различаться и трактовка противоречивых требований. В этом случае ликвидация противоречивости может повлечь за собой «волну изменений» в проектной документации и исходных текстах системы. Для того, чтобы избежать «волн изменений» по завершению инспекций рекомендуется проводить ее поэтапно до начала следующего этапа жизненного цикла или до разработки документов следующего уровня детализации.