Лабораторная работа № 4 Формальная инспекция программного кода (4 час)

Порядок выполнения работы:

1. Изучение теоретических сведений по теме лабораторной работы
2. Проведение формальной инспекции программного модуля.
3. Выбрать объект инспекции
4. Выполнить просмотр кода, все обнаруженные несоответствия должны быть точно локализованы, сформулированы и записаны
5. Оформление отчета по лабораторной работе, который должен содержать:
   - Название и задание к лабораторной работе
   - Ответы на контрольные вопросы и задания
6. Защита лабораторной работы

Формальная инспекция является четко управляемым процессом, структура которого 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.

Как правило, процесс формальной инспекции состоит из пяти фаз: инициализация, планирование, подготовка (экспертиза), обсуждение, завершение.

После устранения обнаруженных в ходе формальной инспекции несоответствий процесс формальной инспекции повторяется, возможно, в другой форме и с другим составом участников.

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.

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.

Рекомендуемая литература

1 Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. СПб.: Питер, 2012.

2 Соммервилл И. Инженерия программного обеспечения: 6-е издание. М.: Вильямс, 2012.

Контрольные задания для СРС

1. Подготовить рабочие материалы, которые включают в себя распечатки инспектируемых документов с пометками и бланки инспекции.
2. Проверить надежность (робастность) программного кода:

1. обработку некорректных данных;
2. проверку диапазонов значений;
3. обработку исключительных ситуаций;

4. наличие сообщений об ошибках.