ПЛАНИРОВАНИЕ РЕВЬЮИРОВАНИЯ

Лекция №3

МДК 03.01. **Моделирование и анализ программного обеспечения**

ПМ.03. Ревьюирование программных продуктов

ЦЕЛИ, КОРРЕКТНОСТЬ И НАПРАВЛЕНИЯ АНАЛИЗА ПРОГРАММНЫХ ПРОДУКТОВ

Лекция №3

МДК 03.01. **Моделирование и анализ программного обеспечения**

ПМ.03. Ревьюирование программных продуктов

ОБЩАЯ ХАРАКТЕРИСТИКА ТЕСТИРОВАНИЯ И ЕГО ЦИКЛ

- **Тести́рование ПО** — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определенным образом (ISO/IEC TR 19759:2005)

- Тестирование представляет собой деятельность по проверке программного кода и документации. Она должна заранее планироваться и систематически проводиться специально назначенным независимым тестировщиком. Работа тестировщика начинается до утверждения спецификаций требований. Он проверяет требования к ПП на полноту и возможность тестирования, определяет методы тестирования
- Одновременно с началом этапа планирования и создания спецификаций требований тестировщик разрабатывает стратегию тестирования. После утверждения спецификаций требований им разрабатывается и детализируется план тестирования. Тогда же тестировщик создает наборы тестов для проведения интеграционного и системного тестирований. Тестирование завершается созданием отчета о тестировании, в котором представляются все результаты его проведения.
- Для каждого программного изделия должен существовать набор тестов, проверяющий его корректность. Существует несколько уровней тестирования, позволяющих полностью проверить программное изделие.

ЦИКЛ ТЕСТИРОВАНИЯ

УРОВНИ ТЕСТИРОВАНИЯ

1. Модульное (осуществляет сам разработчик на этапе разработки);
2. Интеграционное (отвечает независимый тестировщик);
3. Системное (отвечает независимый тестировщик);
4. Выходное (отвечает независимый тестировщик);
5. Приемочное (выполняется совместно с представителями заказчика).

Тестирование 1-4 уровней проводится внутри организации.

Циклом тестирования называется совокупность действий, выполняемых тестировщиком с момента передачи базовой версии ПП тестировщику для интеграционного, системного или приемочного тестирования до момента успешного завершения тестирования.

На каждом проходе цикла тестирования:

- создается базовая версия ПП, подлежащего тестированию
- создается отчет о ходе тестирования
- метрики тестирования (заносятся в базу данных проекта)

ВИДЫ ТЕСТИРОВАНИЯ

I. Модульное тестирование – процесс проверки отдельных программных процедур и подпрограмм, входящих в состав программ или программных систем.

Модульное тестирование производится непосредственным разработчиком и позволяет проверять все внутренние структуры и потоки данных в каждом модуле. Этот вид тестирования является частью этапа разработки. При модульном тестировании выполняется набор тестов, определяемый разработчиком так, чтобы охват тестированием каждого модуля был не менее 70-75%.

Элементами модульного тестирование являются:

- синтаксическая проверка — проверка с использованием некоторого инструментального средства для выявления синтаксических ошибок в программном коде;
- проверка соответствия стандартам кодирования — проверка кода на соответствие стандартам кодирования компании;
- технический обзор программного кода.

После успешного завершения модульного тестирования все измененные модули и наборы тестов сохраняются в БД проекта.

ВИДЫ ТЕСТИРОВАНИЯ (ПРОДОЛЖЕНИЕ)

II. ***Интеграционное тестирование*** – проводится для проверки совместной работы отдельных модулей и предшествует тестированию всей системы как единого целого.

В ходе интеграционного тестирования проверяются связи между модулями, их совместимость и функциональность. Оно осуществляется независимым тестировщиком и входит в состав этапа тестирования.

Элементами интеграционного тестирования являются:

- проверка функциональности — проверка соответствия отдельных функций, выполняемых совокупностями модулей, функциям, заданным в спецификациях требований
- проверка промежуточных результатов — проверка всех промежуточных результатов и файлов на наличие и корректность
- проверка интеграции — проверка того, что модули передают друг другу информацию корректно

Ошибки, выявленные в ходе интеграционного тестирования заносятся в БД ошибок.

Результаты интеграционного тестирования включаются в отчет о ходе тестирования при завершении цикла тестирования.

ВИДЫ ТЕСТИРОВАНИЯ (ПРОДОЛЖЕНИЕ)

III. ***Системное тестирование*** – предназначен для проверки программной системы в целом, ее организации и функционирования на соответствие спецификациям требований заказчика.

Его проводит независимый тестировщик после успешного завершения интеграционного тестирования.

Элементами системного тестирования являются:

- граничное тестирование — тестирование в граничных условиях;
- прогоночное тестирование — тестирование всех функциональных характеристик реальной работы системы;
- целевое тестирование — тестирование на целевой платформе (по возможности)
- проверка документации — проверка пользовательской документации на корректность)
- другие тесты, определяемые тестировщиком

Ошибки, выявленные при системном тестировании, заносятся в БД проекта.

Результаты системного тестирования включаются в отчет о ходе тестирования.

ВИДЫ ТЕСТИРОВАНИЯ (ПРОДОЛЖЕНИЕ)

IV. ***Выходное тестирование*** – завершающий этап тестирования на котором проверяется готовность ПП к поставке заказчику. Данный вид тестирования проводит независимый тестировщик.

Элементами выходного тестирования являются:

- проверка инсталляции — проверка на ясность и корректность инструкций по инсталляции;
- проверка документации — проверка того, что вся необходимая документация полностью подготовлена и готова к передаче заказчику.

Ошибки, выявленные при выходном тестировании, заносятся в БД проекта.

При успешном завершении выходного тестирования ПП поставляется заказчику вместе с отчетом о результатах тестирования.

V. ***Приемочное тестирование*** – проводится организацией, отвечающей за инсталляцию, сопровождение программной системы и обучение конечного пользователя.

ПРОГРАММНЫЕ ОШИБКИ

Одними из распространенных определений программной ошибки являются следующие два:

1) ***программная ошибка*** — это расхождение между программой и ее спецификацией, причем тогда и только тогда, когда спецификация существует и она правильна;
2) ***программная ошибка*** — это ситуация, когда программа не делает того, чего пользователь от нее вполне обоснованно ожидает.

КАТЕГОРИИ ПРОГРАММНЫХ ОШИБОК

Функциональные недостатки – присущи программе, если она не делает того, что должна, выполняет одну из своих функций плохо или не полностью. Функции программы должны быть подробно описаны в ее спецификации, и именно на основе утвержденной спецификации тестировщик строит свою работу.

Недостатки пользовательского интерфейса. Оценить удобство и правильность работы пользовательского интерфейса можно только в процессе работы с ним. Желательно, чтобы в этой работе принимал участие сам пользователь. Этого можно добиться с помощью разработки прототипа ПП, на котором проводятся обкатка и согласование всех требований к пользовательскому интерфейсу с дальнейшей фиксацией их в спецификации требований. После утверждения спецификации требований любые отклонения от нее или невыполнение последних являются ошибкой. Это в полной мере касается и пользовательского интерфейса.

Недостаточная производительность. При разработке некоторого ПП очень важной его характеристикой может оказаться скорость работы, иногда этот критерий задается в требованиях заказчика к ПП. Плохо, если у пользователя создается впечатление, что программа работает медленно, особенно если конкурирующие программы работают ощутимо быстрее, но еще хуже, если программа не удовлетворяет заданным в спецификации требований характеристикам. Это уже ошибка, которая должна быть устранена.

КАТЕГОРИИ ПРОГРАММНЫХ ОШИБОК

- ***Некорректная обработка ошибок.*** Процедуры обработки ошибок — очень важная часть программы. Правильно определив ошибку, программа должна выдать о ней сообщение. Отсутствие такого сообщения является ошибкой в работе программы.
- ***Некорректная обработка граничных условий.*** Существует много различных граничных ситуаций. Любой аспект работы программы к которому применимы понятия «больше» или «меньше», «раньше» или «позже», «первый» или «последний», «короче» или «длиннее», обязательно должен быть проверен на границах диапазона. Внутри диапазонов программа может работать прекрасно, а вот на их границах могут происходить самые неожиданные ситуации, которые, в свою очередь, приводят к ошибкам в работе ПП.
- ***Ошибки вычислений*** – ошибки, вызванные неправильным выбором алгоритма вычислений неправильными формулами, формулами, неприменимыми к обрабатываемым данным. Самыми распространенными среди ошибок вычислений являются ошибки округления.
- ***Ошибки управления потоком.*** По логике работы программы вслед за первым действием должно быть выполнено второе. Если вместо этого выполняется третье или четвертое действие, значит, в управлении потоком допущена ошибка.

КАТЕГОРИИ ПРОГРАММНЫХ ОШИБОК

- ***Ситуация гонок.*** Предположим, в системе ожидаются 2 события: А и Б. Если первым наступит событие А, то выполнение программы продолжится, а если событие Б, то в работе программы произойдет сбой. Разработчики предполагают, что первым всегда должно быть событие А, и не ожидают, что Б может выиграть гонки и наступить раньше. Такова классическая ситуация гонок. Тестировать ситуации гонок довольно сложно. Наиболее типичны они для систем, где параллельно выполняются взаимодействующие процессы и потоки, а также для многопользовательских систем реального времени. Ошибки в таких системах трудно воспроизвести и на их выявление обычно требуется очень много времени.
- ***Перегрузки*** – сбои в работе программы могут происходить из-за нехватки памяти или отсутствия других необходимых системных ресурсов. У каждой программы свои пределы, программа может не справляться с повышенными нагрузками, например, со слишком большими объемами данных. Вопрос в том, соответствуют ли реальные возможности программы и ее требования к ресурсам спецификации программы и как она себя поведет при перегрузках.
- ***Некорректная работа с аппаратурой компьютера***. Программы могут посылать аппаратным устройствам неверные данные, игнорировать их сообщения об ошибках, пытаться использовать устройства, которые заняты или вообще отсутствуют. Даже если нужное устройство просто сломано, программа должна понять это, а не «зависать» при попытке к нему обратиться.

ТЕСТИРОВАНИЕ ДОКУМЕНТАЦИИ

Читая и анализируя документацию, тестировщик прежде всего уделяет внимание ее

- точности,
- полноте,
- ясности,
- простоте использования
- тому, насколько она соответствует ПП.

В ходе тестирования документации наверняка будут найдены проблемы по каждому из указанных критериев. Поэтому заранее следует запланировать многократное тестирование печатного руководства интерактивной справки и других документов.

- Тестировщик, работающий с документацией, отвечает за техническую точность каждого ее слова. Он обязан произвести самую тщательную проверку ее соответствия требованиям спецификации и поведению программы. Особо следует обращать внимание на сложные и запутанные места текста. Они могут отражать неудачно спроектированные элементы самой программы. Технический писатель обязан описать продукт таким, каким он является на самом деле, поэтому помочь устранить запутанные места может только изменение проекта. Настаивать на таких изменениях важно еще и потому, что, в конечном счете, они обеспечат не только простоту документирования продукта, но и легкость его использования.
- Необходимо проверить, не пропущены ли в документации какие-нибудь функции продукта. Технические писатели опираются на спецификацию, собственные заметки и беседы с разработчиками. Разработчики стараются держать их в курсе дела, но иногда забывают сообщить о новых функциях, только что внесенных в программу. Поскольку тестировщики сталкиваются с этими функциями гораздо раньше технических писателей, стоит позаботиться, чтобы их описания попали в документацию. Кроме того, если определенная функция описана в руководстве, то это не значит что она будет описана и в интерактивной справке. Вполне вероятно, что информация легко может потеряться

- Следует помнить, что ***тестировщик одинаково не имеет права требовать изменений как в руководстве к ПП, так и в самом ПП.***

- *Обязанность тестировщика* — выявить проблему, а что с ней делать, решать не ему. В частности, у тестировщика нет никакого права требовать стилистических изменений текста. Он может предложить такие изменения, но технический писатель вправе оставить все как есть и не обязан доказывать тестировщику, что поступает правильно. Для взаимодействия с техническими писателями формальная система отслеживания проблем обычно не применяется.

- Большинство комментариев вносится прямо в копию руководства. По договоренности с техническим писателем выбирается способ выделения в тексте правок и комментариев. Копии комментариев следует сохранять и проверять по ним очередные версии документации.