**Документирование и жизненный цикл дефекта.**

**Документирование дефекта** заключается в том, что каждый дефект, обнаруженный в процессе тестирования, заносится в базу дефектов. Для этого используют специализированные базы, поддерживающие хранение и отслеживание дефектов. При занесении нового дефекта указывают, как минимум, следующую информацию:

наименование подсистемы, в которой обнаружен дефект;

версию продукта, на которой дефект был найден;

описание дефекта;

описание процедуры (шагов, необходимых для воспроизведения дефекта);

номер теста, на котором дефект был обнаружен;

уровень дефекта, то есть степень его серьёзности с точки зрения критериев качества продукта или заказчика.

**Жизненный цикл дефекта** — это набор определённых этапов, через которые проходит дефект от выявления до устранения. Занесённый в базу дефектов новый дефект находится в состоянии «New». После того, как команда разработчиков проанализирует дефект, он переводится в состояние «Open» с указанием конкретного разработчика, ответственного за исправление дефекта. После исправления дефект переводится разработчиком в состояние «Resolved». При этом разработчик должен указать следующую информацию:

причину возникновения дефекта;

место исправления, как минимум, с точностью до исправленного файла;

краткое описание того, что было исправлено;

время, затраченное на исправление.

**Документирование и жизненный цикл дефекта** - это важные процессы в управлении качеством программного обеспечения. Давайте рассмотрим основные этапы жизненного цикла дефекта:

Обнаружение дефекта: В этом этапе дефект обнаруживается пользователями, тестировщиками или разработчиками. Он может быть выявлен в процессе тестирования или эксплуатации системы.

Регистрация дефекта: Обнаруженный дефект регистрируется в системе отслеживания дефектов (например, Jira, Bugzilla). Важно указать все необходимые данные, такие как описание дефекта, шаги воспроизведения, ожидаемый и фактический результат, а также приоритет и серьезность дефекта.

Анализ дефекта: На этом этапе дефект анализируется командой разработчиков и тестировщиков для определения причины проблемы и возможных решений. Важно понять, как дефект влияет на систему и какие изменения необходимы для его исправления.

Исправление дефекта: Разработчики вносят изменения в код для исправления дефекта. После этого код проходит тестирование, чтобы убедиться, что дефект был исправлен и не вызвал новых проблем.

Повторное тестирование: После исправления дефект проходит повторное тестирование, чтобы убедиться, что он действительно был исправлен и система работает правильно. Это может включать регрессионное тестирование, чтобы проверить, не возникли ли новые дефекты.

Закрытие дефекта: Если дефект успешно исправлен и повторное тестирование прошло без проблем, дефект может быть закрыт в системе отслеживания дефектов. Важно задокументировать все действия, связанные с исправлением дефекта, чтобы иметь историю и возможность последующего анализа.

\_\_\_\_\_

Документирование и управление дефектами являются важной частью процесса разработки программного обеспечения, так как они помогают отслеживать и устранять ошибки на всех этапах проекта. Давайте рассмотрим, как документирование дефектов и их жизненный цикл могут быть организованы.

**Этапы жизненного цикла дефекта**

Идентификация: Обнаружение дефекта тестировщиком или пользователем.

Регистрация: Запись дефекта в систему отслеживания дефектов с указанием подробностей (идентификатор, описание, шаги для воспроизведения, ожидаемые и фактические результаты, приоритет, серьезность, окружение).

Классификация: Оценка дефекта и присвоение ему приоритета и серьезности.

Назначение: Назначение дефекта разработчику или команде для анализа и исправления.

Анализ и исправление: Разработчик анализирует причину дефекта и вносит изменения в код для его исправления.

Тестирование исправления: Тестировщик проверяет исправление и повторно тестирует систему, чтобы убедиться, что дефект устранен и не повлиял на другие функции (ретестирование и регрессионное тестирование).

Закрытие: Если дефект устранен, он закрывается в системе отслеживания дефектов. Если он не устранен, процесс возвращается к этапу "Анализ и исправление".

Анализ дефектов: Анализируется история дефекта для выявления причин и предотвращения подобных проблем в будущем (например, анализ первопричин).

**Документирование дефектов**

Идентификатор дефекта: Уникальный номер или код для каждого дефекта.

Название дефекта: Краткое описание проблемы.

Описание: Подробное описание проблемы, включая шаги для воспроизведения, ожидаемые и фактические результаты.

Шаги для воспроизведения: Подробная инструкция для воспроизведения дефекта.

Окружение: Информация о программной и аппаратной среде, в которой был обнаружен дефект.

Приоритет: Оценка важности и срочности исправления дефекта.

Серьезность: Оценка влияния дефекта на систему.

Ответственный: Имя разработчика или команды, назначенной для исправления дефекта.

Статус: Текущий статус дефекта (открыт, в работе, исправлен, закрыт).

Комментарии: Дополнительные заметки и комментарии по поводу дефекта и его исправления.

**Примеры систем отслеживания дефектов**

JIRA: Популярный инструмент для управления проектами и отслеживания дефектов, предоставляющий гибкие возможности настройки и интеграции.

Bugzilla: Открытый инструмент для отслеживания дефектов, используемый многими крупными проектами с открытым исходным кодом.

Redmine: Инструмент управления проектами с функциями отслеживания дефектов и поддержкой множества плагинов.

Trello: Инструмент для управления задачами, который также может использоваться для отслеживания дефектов с помощью досок и карточек.

\_\_\_\_\_

Документирование и жизненный цикл дефекта включает в себя процесс регистрации, отслеживания и управления дефектами, найденными в программном продукте во время тестирования или эксплуатации. Жизненный цикл дефекта обычно включает следующие этапы:

Регистрация дефекта: дефект регистрируется в системе управления дефектами, где указываются его описание, приоритет, статус и другие атрибуты.

Отслеживание и исправление дефекта: дефект назначается разработчику для исправления, и его статус отслеживается. Когда дефект исправлен, он проверяется тестировщиком.

Подтверждение исправления дефекта: если дефект исправлен, тестировщик проводит повторное тестирование, чтобы убедиться, что дефект действительно исправлен.

Закрытие дефекта: если дефект более не обнаруживается и считается исправленным, он закрывается в системе управления дефектами.

\_\_\_\_\_

**Разработать модульные тесты для мобильного приложения Calculate, выполнить тестирование интерфейса. Приложение находится «Resorce/Calc/». Оформить тест-кейсы для модульного тестирования.**