

## ТЕХНИЧЕСКИЕ НАУКИ

***Усманова Илюся Рамилевна***

инженер по тестированию ПО

ICL-КПО ВС

магистрантка

ФГАОУ ВПО «Казанский (Приволжский)

федеральный университет»

г. Казань, Республика Татарстан

***Вафина Рашидя Фуатовна***

старший преподаватель

Институт международных отношений,

истории и востоковедения

ФГАОУ ВПО «Казанский (Приволжский)

федеральный университет»

г. Казань, Республика Татарстан

### **КАЧЕСТВО И ТЕСТИРОВАНИЕ ПРОЕКТА:**

#### **ВЕДУЩИЕ СТРАТЕГИИ**

*Аннотация:* статья посвящена обзору жизненного цикла проекта с углублением в область качества и тестирования ПО. Изучение роли качества в жизненном цикле проекта рассматривается как один из важнейших аспектов, кроме этого в статье представлены стратегии, уровни тестирования (модульный, интеграционный и системный) и виды тестирования (по объекту тестирования, по знанию системы, по времени тестирования).

*Ключевые слова:* жизненный цикл проекта, характеристики качества, программное обеспечение, уровни тестирования ПО.

Поскольку информационные технологии все больше затрагивают нашу жизнь, ошибки программного обеспечения могут привести к нанесению материального ущерба, нарушению секретности и к многому другому. Следовательно,

его высокое качество воспринимается как обязательное условие его применения. Для того чтобы меньше возникало ошибок в ходе работы с программным обеспечением, нужно измерять качество ПО на протяжении всего жизненного цикла проекта. В большинстве случаев, качество ПО достигается его тестированием.

При рассмотрении общей структуры жизненного цикла проекта важно выделить фазы проекта: начальная и финальная. На начальной фазе определяется общая идея. Сформировав идею, в конце инициации на выходе получается устав. Так же на начальной фазе формируется команда управления проектом. С момента запуска и до остановки проекта на выходе следующее: устав, описание содержания, план, базовый план, прогресс, приемка, одобрение и передача заказчику продукта. Результатом является конечный продукт.

Структура жизненного цикла проекта включает в себя:

- анализ требований;
- разработку дизайна ПО;
- разработку ПО;
- тестирование и документирование;
- эксплуатацию и поддержку.

Особое внимание мы уделяем четвертому этапу жизненного цикла, на котором проводится тест. Данный процесс напрямую связан с качеством ПО, т.к. оно контролируется и обеспечивается в процессе тестирования и должно соответствовать требованиям и ожиданиям заказчика. Качество продукта характеризуют его функциональность, надежность, практичность, эффективность, сопровождаемость и мобильность.

Функционал должен быть пригодным, точным для получения данных, уметь взаимодействовать с другими компонентами, быть безопасным и соответствовать функциональным требованиям заказчика.

Надежность ПО характеризуют нефункциональные требования. Нефункциональные требования описывают как должна работать система или программный продукт, и какими свойствами или характеристиками она должна обладать [1].

Продукт должен быть зрелым, т.е. с наименьшей вероятностью появления ошибок. Это достигается с помощью большого количества выпуска программы. Более зрелая система больше устойчива к сбоям. Чтобы система была устойчива к сбоям, должны фиксироваться данные о запросах пользователей. А также, немаловажный фактор, восстанавливаемость системы. Должна быть возможность восстановления программы. Кроме всех характеристик к надежности ПО, продукт должен соответствовать параметрам надежности, которые описываются в требованиях к надежности.

Практичность ПО также относится к нефункциональным требованиям. Под практичностью подразумевается взаимодействие пользователя с системой с помощью интерфейса. Продукт должен быть понятен. Понятность – это обучение пользователя работать с системой и поддержание процесса работы пользователя. Кроме понятности, продукт должен быть пригодным к эксплуатации. Под пригодностью к эксплуатации подразумеваются требования к оборудованию и поддержка его характеристик. Первое на что обращает внимание пользователь – это дизайн, красота продукта. Это пункт привлекательности. Если даже продукт привлекателен и его неудобно использовать, то пользователь ненадолго задержится с этой системой. Кроме привлекательности должно быть удобно его использовать.

Эффективность ПО достигается быстродействием, возможностью использования ресурсов и соответствием ожиданиям эффективности. По работе быстродействия можно привести пример с проблемой конфигурации оборудования – это возможность влияния на соединение с базой данных или замена жесткого диска. Кроме быстродействия должно быть использование ресурсов в виде памяти или процессора. Чтобы соответствовать ожиданиям эффективности, нужно выяснять у заказчика чего именно он желает.

Значимым является придерживаться четырех главных принципов сопровождаемости ПО:

- анализируемость;
- изменяемость;

- стабильность;
- тестируемость.

Во-первых, должна быть возможность просмотра событий о запросах пользователей. Во-вторых, возможность быстро изменить систему любым способом. В-третьих, продукт должен быть стабилен. И наконец, должна быть документация, где есть описание тестирования, и пригодность к поддержке.

Также, продукт характеризует мобильность. Это возможность ПО к адаптации к разным платформам. Чем меньше требований к оборудованию, тем лучше адаптируемо ПО. Должна быть легкая установка, возможность работы с другими системами, а также заменяемость, т.е. возможность заменить аналогами, и портируемость (перенос).

Качество ПО планируется определением стратегии тестирования и критериев приемочных испытаний. Обеспечение качества достигается написанием сценариев использования, которые описывают поведение системы. И наконец, процесс контроля качества происходит в виде тестирования.

Тестирование ПО – это проверка программного продукта на соответствие требованиям заказчика. Цель тестирования заключается в том, чтобы показать разработчику и заказчику, что продукт соответствует требованиям, и выявить ситуации, где программа работает неправильно или не по спецификациям.

К процессу тестирования ПО относятся: определение стратегии тестирования, разработка плана тестирования, написание тест кейсов, проведение тестирования, написание отчета о тестировании.

Стратегия описывает каким образом будет тестироваться ПО. Это отражается уровнями тестирования. Уровни тестирования ПО бывают в виде модульного, интеграционного и системного тестирования. Модульное тестирование означает, что разработчик сам проверяет бизнес логику функционала с помощью unit тестов, которые позволяют проверить на корректность отдельные модули исходного кода программы [2]. Но, если разработчик работает как разработчиком, как и тестировщиком, то он должен тратить на тестирование 30% своего времени. Интеграционное тестирование позволяет проверить взаимодействие

компонент между собой. Системное тестирование покрывает тестированием всю систему.

В плане тестирования описывается весь объем работ по тестированию – это кто, что, каким образом и какими видами будет проводиться тестирование.

Тест кейсы описывают шаги прохождения тестирования и ожидаемый результат.

Само тестирование может проводится как вручную, так и автоматизированно. Ручное тестирование производится тестировщиком без использования программных средств. В автоматизированном тестировании используются программные средства для выполнения тестов и проверки результатов выполнения, что помогает сократить время тестирования и упростить его процесс [3], что значительно повышает эффективность тестировщика.

В отчете о тестировании содержится три основных пункта:

- что было сделано;
- сколько проблем было выявлено;
- результаты обработки ранее выявленных проблем.

Как было сказано ранее, план тестирования содержит виды тестирования. Выделяют 3 категории видов тестирования: по объекту тестирования, по знанию системы, по времени тестирования.

Виды тестирования по объекту тестирования:

- 1) функциональное тестирование – тестирование функционала системы;
- 2) интеграционное тестирование – проверка системы в целом;
- 3) тестирование производительности:
  - нагрузочное – проверка нагрузки от средней до пиковой;
  - стресс-тестирование – проверка при пиковой нагрузке;
  - тестирование стабильности – по длительности;
  - конфигурационное тестирование – проверка параметров оборудования.
- 4) юзабилити тестирование – интерфейс;
- 5) тестирование пользовательского интерфейса – удобство;

б) тестирование безопасности – тестирование входа в систему, несанкционированного портирования, доступа к данным, шифрования;

7) тестирование локализации – языки интерфейса;

8) тестирование совместимости – совместимость с платформой, с другими приложениями;

9) регрессионное тестирование – проверка ранее созданного функционала.

Выделяют 3 вида тестирования по знанию системы: черный, белый и серый ящик. Тестирование методом черного ящика означает тестирование без знания внутренностей системы, т.е. без программного кода, без постановки. Тестирование методом белого ящика подразумевает, что тестирующий знает, что будет. В тестирование методом серого ящика включен как черный, так и белый ящик. Кроме тестирования по знанию системы, также еще есть тестирование по времени.

Тестирование по времени тестирования включает альфа и бета тестирование. Альфа тестирование проводится в целевой аудитории, т.е. тестирует специальная команда тестирующих, а бета тестирование проводится пользователями.

Чтобы не возникали случаи ошибок и сбоев в программном обеспечении, обязательно вводится процесс качества ПО в жизненный цикл проекта. Качество оказывает существенное влияние на продукт, поэтому оно играет важную роль в жизненном цикле. В заключении, важно отметить, что при обеспечении качества ПО необходимо отметить основные правила: не нужно делать больше, чем требуется; нельзя сделать меньше, чем требуется; важно сделать точно, что требуется.

### ***Список литературы***

1. Нефункциональные требования к программному обеспечению [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/231961/>

2. Модульное тестирование [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Модульное\\_тестирование](https://ru.wikipedia.org/wiki/Модульное_тестирование)

3. Автоматизированное тестирование [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Автоматизированное\\_тестирование](https://ru.wikipedia.org/wiki/Автоматизированное_тестирование)