

Министерство образования и науки РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра материаловедения,
технологии и управления качеством

ОБЕСПЕЧЕНИЕ КАЧЕСТВА ПРОГРАММНЫХ СРЕДСТВ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 537 группы
направления 27.03.02 «Управление качеством»
факультета нано - и биомедицинских технологий
Горявина Евгения Викторовича

Научный руководитель

доцент, к.ф.-м.н. _____ А.Э. Постельга

подпись, дата

Зав. кафедрой

профессор, д.ф.-м.н. _____ С.Б. Вениг

подпись, дата

Саратов 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Методы обеспечения качества при проектировании программных средств.....	4
1.1 Анализ характеристики качества программных средств.....	4
1.2 Методы улучшения качества кода.....	5
2 Проектирование процесса тестирования ПО.....	6
2.1 Жизненный цикл тестирования.....	8
2.2 Классификация тестирования.....	8
3 Автоматизация тестирования ПО.....	9
3.1 Выгода и риски автоматизации тестирования.....	9
3.2 Области применения автоматизации.....	11
ЗАКЛЮЧЕНИЕ.....	12

ВВЕДЕНИЕ

Цель работы: определение оптимального подхода к построению рабочего процесса обеспечения качества и тестирования ПО.

Для достижения поставленной цели необходимо решить следующие задачи:

- систематизация, закрепление и углубление теоретических знаний по направлению и умений применять их для решения конкретных практических задач;
- применение навыков ведения самостоятельной научно-исследовательской работы и использование соответствующих методик для решения конкретных задач;
- обобщение и оценка существующих технологий и процессов, касательно проблематике мануального и автоматизированного тестирования;
- проведение дополнительных исследований передовых отечественных и зарубежных источников в области качества;
- оценка применения современных подходов на практике и формирование выводов.

1. Методы обеспечения качества при проектировании программных систем

Согласно ISO 8402 качество программного обеспечения – совокупность характеристик ПО, относящихся к его способности удовлетворять установленные и предполагаемые потребности. Первоочередная задача – определение требований к программному продукту: функциональные и нефункциональные требования.

Ошибка в толковании требований на ранних этапах может привести к значительным затратам при доработке готового продукта.

1.1 Анализ характеристики качества программных систем

Качество программного обеспечения состоит из трех аспектов:

- качество процесса;
- качество продукта;
- качество сопровождения.

Аспекты качества программных средств схематично изображены на рисунке 1.

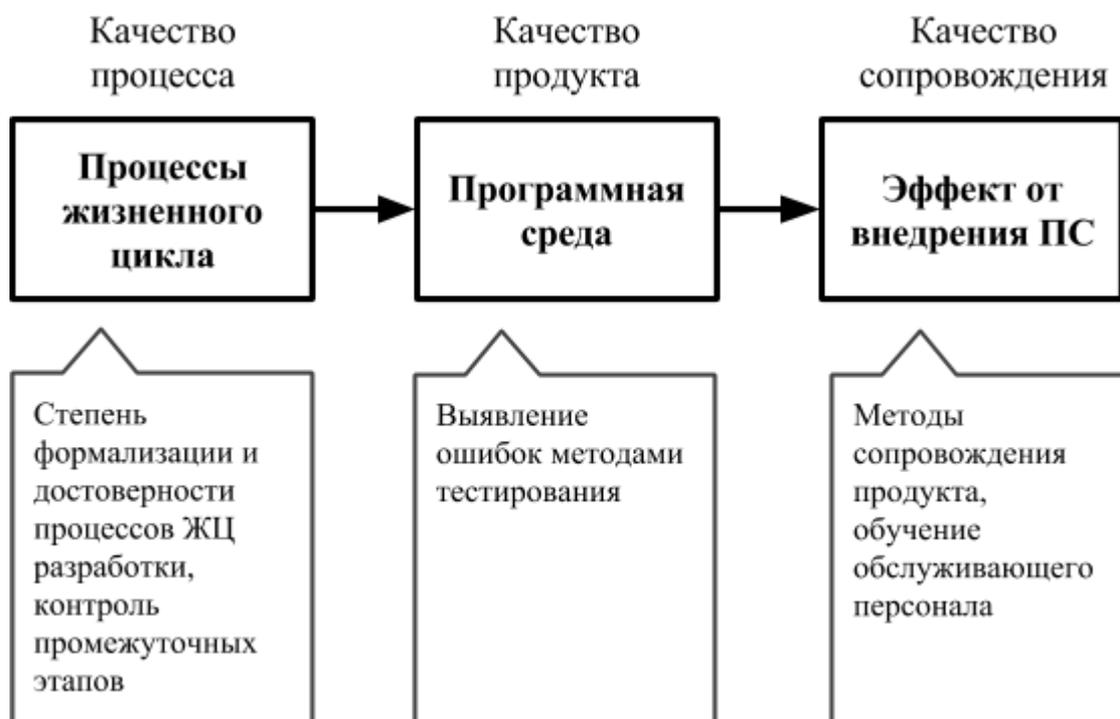


Рисунок 1– Аспекты качества программных средств

Далее определяются характеристики качества, отражающие различные точки зрения конечного пользователя на качество. Схема представлена на рисунке 2.



Рисунок 2 – Обобщенная модель качества программных средств.

1.2 Методы улучшения качества кода

Общепринятые концепции совершенствования качества, такие как предотвращение и ранняя диагностика ошибок, постоянное совершенствование (continuous improvement) и внимание к требованиям заказчика (customer focus), применимы и к программной инженерии.

Наиболее известными подходами к улучшению качества являются TQM (Total Quality Management), PDCA (Plan, Do, Check, Act), итеративная разработка. Немаловажным условием успешного выполнения процессов, оценки продуктов и получения всех необходимых данных является поддержка со стороны менеджмента.

SQM (Software Quality Management) – управление качеством программного обеспечения, SQM применяется ко всем процессам и их аспектам, сопровождающим создание программного обеспечения, а именно: к

требованиям, к владельцам процессов, к изменениям процессов, к результатам этих изменений. Процессы управления качеством ПО состоят из большого числа пунктов. Главной их целью является преждевременное нахождение ошибок и дефектов создаваемой системы. В одних случаях эти процессы помогают найти сам дефект напрямую, в других – всего лишь указать направление дальнейших исследований.

2 Проектирование процесса тестирования ПО

Задача данного раздела – познакомить с существующими методами тестирования, провести анализ и выработку оптимального подхода к построению рабочего процесса по обеспечению качества ПО.

Тестирование программного обеспечения – процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта.

На протяжении десятилетий развития разработки ПО к вопросам тестирования и обеспечения качества подходили по-разному. Выделим основные ретроспективные этапы “эпох тестирования”. Наглядное отображение представлено на рисунке 3.

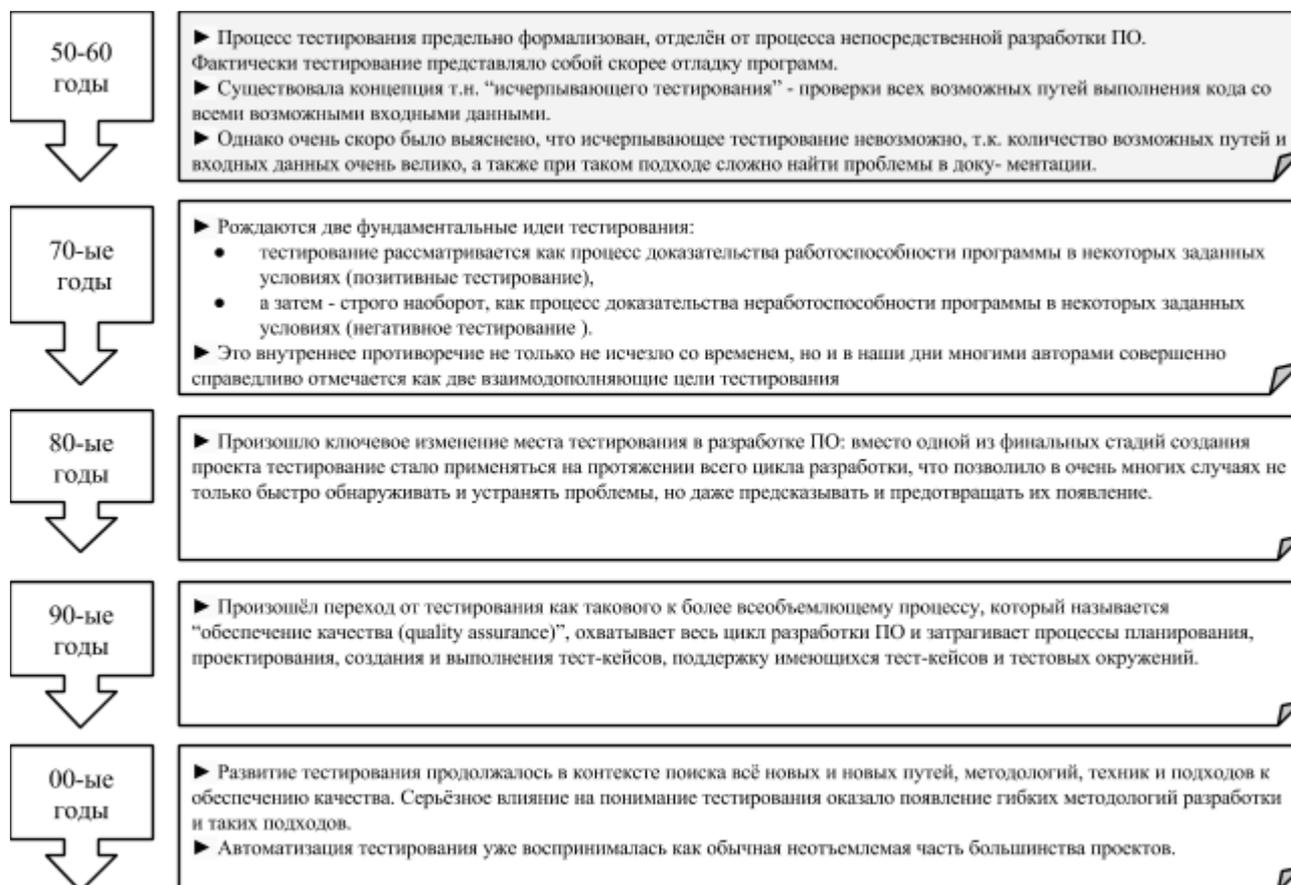


Рисунок 3 – Основные этапы развития тестирования

О современном этапе развития тестирования мы будем говорить на протяжении всей остальной работы. Если же отметить основные характеристики, присущее современному этапу, то получится примерно такой список:

- гибкие методологии и гибкое тестирование;
- глубокая интеграция с процессом разработки;
- широкое использование автоматизации;
- колоссальный набор технологий и инструментальных средств;
- кроссфункциональность команды (когда тестировщик и программист во многом могут выполнять работу друг друга).

2.1 Жизненный цикл тестирования

Если следовать общей логике итеративности, преобладающей во всех современных моделях разработки ПО, жизненный цикл тестирования можно выразить замкнутой последовательностью действий, схематично изображённой на рисунке 4.

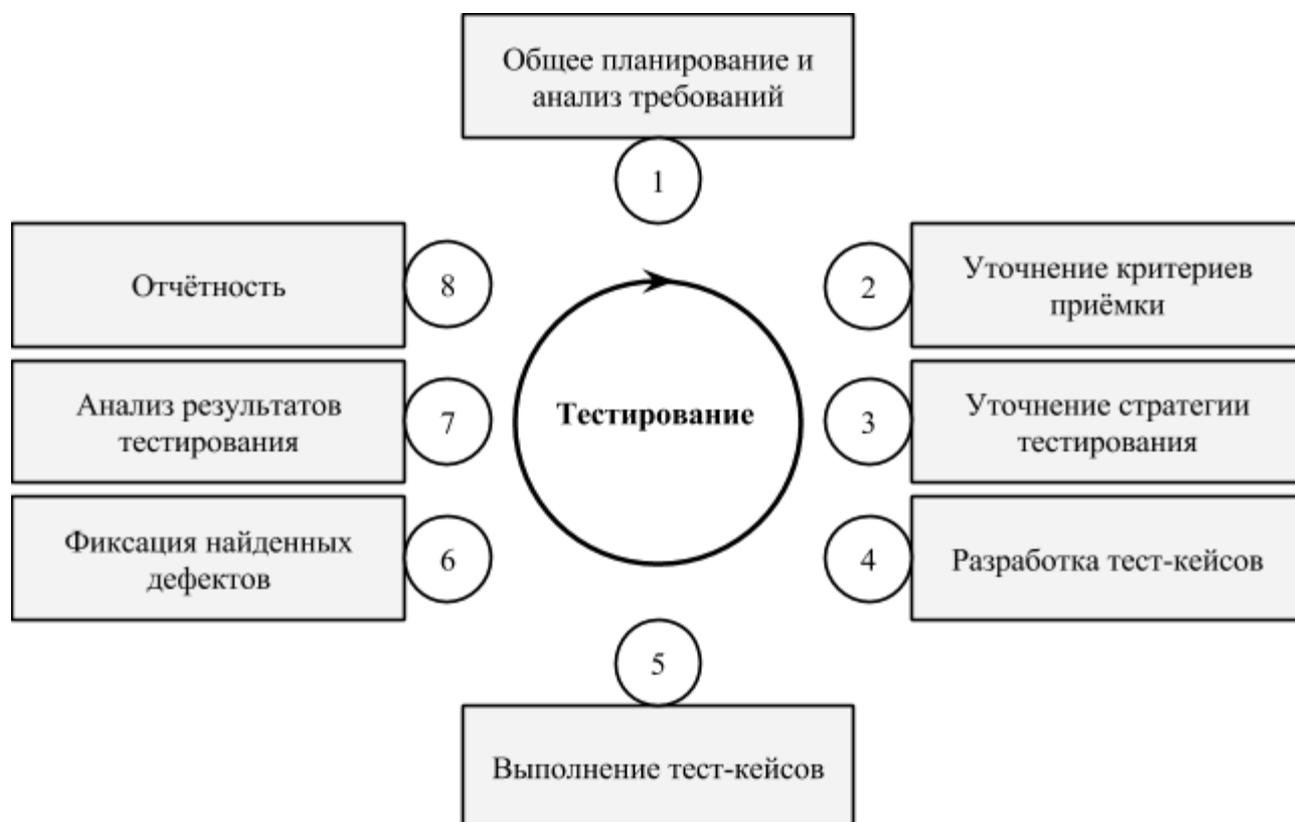


Рисунок 4 – Жизненный цикл тестирования

2.2 Классификация тестирования

Тестирование можно классифицировать по очень большому количеству признаков. Нижеприведённый список представляет из себя краткую модель, схематично изображённую на рисунке 8. Итак, тестирование можно классифицировать:



Рисунок 8 – Упрощённая классификация тестирования

3. Автоматизация тестирования ПО

Автоматизированное тестирование – это набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования.

3.1 Выгоды и риски автоматизации тестирования

Если выразить все преимущества и недостатки автоматизации тестирования одной фразой, то получается, что автоматизация позволяет ощутимо увеличить тестовое покрытие, но при этом столь же ощутимо увеличивает риски.

С использованием автоматизации мы получаем возможность увеличить тестовое покрытие¹ за счёт:

- выполнения тест-кейсов, о которых раньше не стоило и думать;
- многократного повторения тест-кейсов с разными входными данными;
- высвобождения времени на создание новых тест-кейсов.

¹ Coverage, Test coverage. The degree, expressed as a percentage, to which a specified coverage item has been exercised by a test suite. [ISTQB Glossary]

Покрытие (coverage) – процентное выражение степени, в которой исследуемый элемент (coverage item²) затронут соответствующим набором тест-кейсов.

Самыми простыми представителями метрик покрытия можно считать:

- Метрику покрытия требований (требование считается “покрытым”, если на него ссылается хотя бы один тест-кейс):

$$R^{SimpleCoverage} = \frac{RCovered}{RTotal} \cdot 100\%, \quad (1)$$

где $R^{SimpleCoverage}$ — метрика покрытия требований,

$RCovered$ — количество требований, покрытых хотя бы одним тест-кейсом,

$RTotal$ — общее количество требований.

- Метрику плотности покрытия требований (учитывается, сколько тест-кейсов ссылается на несколько требований):

$$R^{DensityCoverage} = \frac{\sum Ti}{TTotal \times RTotal} \cdot 100\%, \quad (2)$$

где $R^{DensityCoverage}$ — плотность покрытия требований,

Ti — количество тест-кейсов, покрывающих i -е требование,

$TTotal$ — общее количество тест-кейсов,

$RTotal$ — общее количество требований.

² Coverage item. An entity or property used as a basis for test coverage, e.g. equivalence partitions or code statements. [ISTQB Glossary]

Но с автоматизацией не все так гладко. Наглядно одну из серьёзных проблем можно представить на рисунке 9:

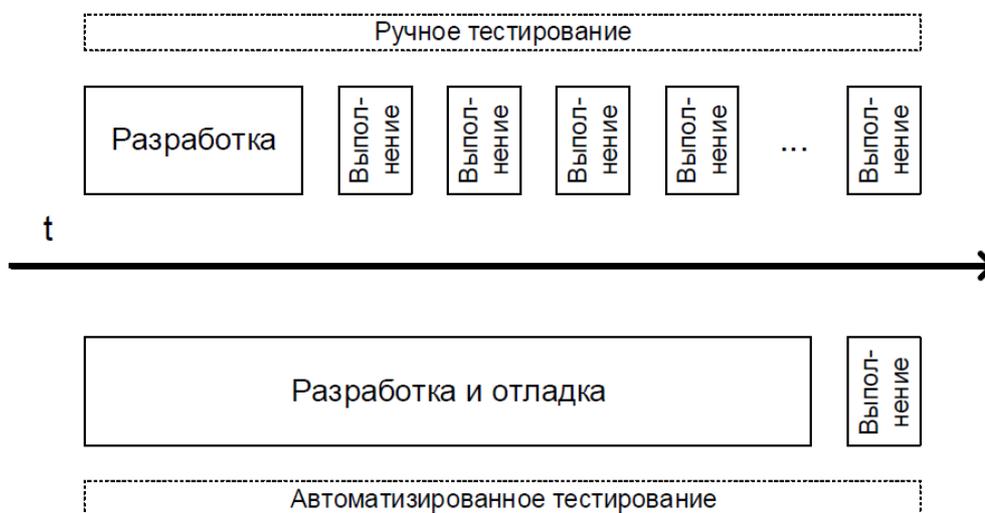


Рисунок 9 – Соотношение времени разработки и выполнения тест-кейсов в ручном и автоматизированном тестировании

3.2 Области применения автоматизации

Рассмотрим список задач, решить которые помогает автоматизация:

- Выполнение тест-кейсов, непосильных человеку.
- Решение рутинных задач.
- Ускорение выполнения тестирования.
- Высвобождение человеческих ресурсов для интеллектуальной работы.
- Увеличение тестового покрытия.
- Улучшение кода за счёт увеличения тестового покрытия и применения специальных техник автоматизации.

Вывод: стоит помнить, что эффект от автоматизации наступает не сразу и не всегда. Как и любой дорогостоящий инструмент, автоматизация при верном применении может дать ощутимую выгоду, но при неверном принесёт лишь весьма ощутимые затраты.

ЗАКЛЮЧЕНИЕ

В ходе выполнения настоящей работы была выполнена систематизация, закрепление и углубление теоретических знаний по направлению с дальнейшим применением их для решения конкретных практических задач.

Применены навыки ведения самостоятельной научно-исследовательской работы и использование соответствующих методик для решения конкретных задач.

Произведены обобщение и оценка существующих технологий и процессов, касательно проблематики мануального и автоматизированного тестирования в рамках проведения дополнительных исследований передовых отечественных и зарубежных источников в области качества, а также оценка применения современных подходов на практике и формирование выводов.

Таким образом, можно утверждать, что цель настоящей работы была достигнута.