

СПИСОК ЛИТЕРАТУРЫ

1. *Дериземля В.А.* Современное состояние и перспективы использования низкопотенциального тепла в народном хозяйстве (обзорная статья) // Молодой ученый. – 2017. – № 2(136). – С. 106-110.
2. *Рыбач Л.* Состояние и перспективы геотермальных тепловых насосов (ГТН) в Европе и мире; аспекты устойчивости GHP. Международный курс геотермальных тепловых насосов, 2002 г.
3. *Ширяев Р.С.* Низкопотенциальное тепло Земли как источник теплоснабжения, 2016. – URL:https://aqua-therm.ru/netcat_files/387/613/SPEZ_EXPERT_5_53_2016.pdf (дата обращения: 19.10.2022).

ANALYSIS OF THE USE OF LOW-POTENTIAL ENERGY OF THE SOIL**ZENINA Anastasia Romanovna**

student

PAVLENKO Anastasia Vyacheslavovna

student

Don State Technical University

Rostov-on-Don, Russia

An urgent problem at enterprises is the efficient use of energy resources. A productive direction seems to be the use of heat pump systems of heat supply, using the ground of the surface layers of the earth as a source of low-potential heat. Systems of using low-potential heat energy of the earth are divided into open and closed systems. Low-potential energy of the Earth has a number of advantages, representing a reliable source of energy, which can be used everywhere.

Key words: low-potential sources, energy potential, heating systems, ground energy, environmental friendliness.

АНАЛИЗ МЕТОДОВ ТЕСТ-ДИЗАЙНА В ТЕСТИРОВАНИИ ПРОГРАММНОГО ПРОДУКТА**ПРОКАЗОВА Жанна Витальевна**

магистрант

ФГБОУ ВО «Донской государственный технический университет»

г. Ростов-на-Дону, Россия

Проверка качества разрабатываемого программного обеспечения – это важнейшая фаза жизненного цикла разработки. В результате высокой конкуренции на рынке качественный продукт играет решающую роль при выборе клиента. Из-за этих условий в команде разработки и появилась такая роль как тестировщик программного обеспечения.

Ключевые слова: тестирование, тест-дизайн, качество, проверка, программный продукт.

В настоящее время набирает популярность такая профессия как тестировщик программного обеспечения. Это относительно молодая профессия, она появилась в процессе развития разработки программного обеспечения. Сейчас практически во всех командах есть тестировщик или отдел тести-

рования. Тестирование также развивалось в процессе развития информационных технологий. Из ручного тестирования появилось автоматизированное тестирование, что освободило тестировщиков от рутинной работы – проверки измененного функционала программного обеспечения (регрессионное тестирование).

Регрессионное тестирование представляет собой вид тестирования, которое заключается в проверке того, что внесенное изменение кода программного обеспечения не оказало влияния на неизменный функционал программного обеспечения.

В процессе работы тестировщика было выявлено 7 принципов, которые позволяют рационально отнестись к процессу тестирования. Выделяют следующие принципы:

1. Процесс проведения тестирования показывает только наличие дефектов, но не их отсутствия. Нельзя уверенно утверждать, что дефектов нет, они есть практически во всех проектах, но незначительные.

2. Невозможно провести полное тестирование, это невыполнимо. Исчерпывающее тестирование с применением всех возможных вариантов ввода и комбинаций недостижимо, но кроме тривиальных случаев.

3. Необходимо проводить тестирование на ранних этапах разработки программного обеспечения, чтобы исключить появление дефектов в технической документации до этапа кодирования.

4. Дефекты имеют свойства скапливаться в одном месте (модули). В разных модулях программного обеспечения содержится различное количество дефектов, то есть плотность дефектов в разных местах может отличаться.

5. Парадокс пестицида. Если всегда проводить одни и те же тест-кейсы, в итоге данный набор не сможет находить новые дефекты. Во избежание данной ситуации необходимо периодически обновлять тестовые сценарии.

6. Тестирование напрямую зависит от контекста. Для тестирования разных программных обеспечений необходимо применять разные подходы в зависимости от контекста описания технической документации.

7. Заблуждение, что ошибок нет. Отсутствие дефектов при тестировании не говорит об идеальном программном обеспечении. Главным критерием готовности продукта является удобство пользования.

Для обеспечения максимального покрытия тестовыми сценариями в процессе тестирования были разработаны методы (техники тест-дизайна), чтобы при меньших затратах покрыть большую часть функционала программного обеспечения. Разберем и сравним тест-дизайн:

1. Классы эквивалентности. Данный метод дает возможность уменьшить количество тестовых сценариев. С помощью анализа предметной области объекта тестирования берется одно значение из созданного класса и подразумевается, что для всех значений в этом классе поведение будет одинаковым.

2. Анализ граничных значений. Данный метод немного похож на разделение на классы эквивалентности, но основное отличие между этими двумя методами заключается в том, что берутся значения на границе между выбранными классами.

3. Диаграмма переходов и состояний. Данный метод наглядно демонстрирует состояние программного обеспечения при разных вводных параметрах на разных этапах работы. При визуализации тестовых сценариев процесс тестирования выглядит нагляднее, воспринимать проще, чем текст.

4. Парное тестирование. Данный метод наиболее сложный. В основе метода лежит комбинирование вводных значений. Процесс комбинирования дает возможность создать уникальные пары, тем самым позволяя тестировать программное обеспечение при наличии данных в разных сочетаниях.

По отдельности данные методы сокращают время для проведения тестирования программного обеспечения с помощью уменьшения возможных вводных данных. Наиболее лучшим вариантом будет комбинирование методов, это намного сократит время, и будет наиболее эффективно. Но нельзя забывать, что 100% покрытие тестовыми сценариями программного обеспечения практически невозможно добиться.

Но при попытке достичь 100% покрытия тестами программного обеспечения может потребоваться огромное количество времени. В большинстве случаев это не стоит потраченного времени.

Таким образом, можно сделать вывод, что тестирование увеличивает знание о качестве программного продукта, тем самым позволяя увеличить это качество в случае несоответствия. А техники тест-дизайна дают возможность при сокращении времени на проведение тестирования увеличить покрытие тестовыми сценариями, тем самым за более короткий промежуток времени проверить большую часть программного обеспечения.

СПИСОК ЛИТЕРАТУРЫ

1. *Короткий Д.А.* Автоматизация тестирования программного обеспечения автоматизированных комплексов скользящей опалубки // Вестник Донского государственного технического университета. – 2012. – № 12(3). – С. 37-43.
2. Немного о простом. Тест-дизайн. Часть 1. 2022. – URL:<https://habr.com/ru/post/462553/> (дата обращения: 17.10.2022).
3. Теоретические основы тестирования. 2022. – URL:<https://clck.ru/UMAEj> (дата обращения: 17.10.2022).
4. Тестирование программного обеспечения. 2022. – URL:<https://clck.ru/umaf4> (дата обращения: 17.10.2022)
5. Тест-дизайн и техники тест-дизайна (Test Design and Software Testing Techniques). 2022. – URL:https://vladislavremeev.gitbook.io/qa_bible/test-dizain/test-dizain-i-tekhniki-test-dizaina-test-design-and-software-testing-techniques (дата обращения: 17.10.2022).

ANALYSIS OF TEST-DESIGN METHODS IN SOFTWARE PRODUCT TESTING

PROKAZOVA Zhanna Vitalyevna

master's student

Don State Technical University

Rostov-on-Don, Russia

Checking the quality of the software being developed is the most important phase of the development life cycle. As a result of high competition on the market, a quality product plays a decisive role in customer selection. Because of these conditions, such a role as software tester has emerged in the development team.

Key words: testing, test design, quality, verification, software product.

БИЗНЕС-АНАЛИТИКА В РАЗРАБОТКЕ ПРОГРАММНОГО ПРОДУКТА

ПРОКАЗОВА Жанна Витальевна

магистрант

ФГБОУ ВО «Донской государственный технический университет»

г. Ростов-на-Дону, Россия

В современном мире бизнес-анализ представляет собой одну из услуг, которой зачастую не предоставляют должного внимания при разработке программного продукта. Множество проектов прогорели из-за плохого построения бизнес-процессов и нечеткого анализа рисков проекта.

Ключевые слова: аналитика, разработка, методология, программный продукт, бизнес.

Любой проект начинается с фундамента, а именно с описания бизнес-процессов. В процессе разработки программного продукта описание функционала играет большую роль. Процесс разработки программного продукта можно сравнить со строитель-

ством здания. Нельзя начать строить стены, пока не заложен фундамент.

В процессе разработки программного продукта фундамент закладывает именно бизнес-аналитика. Даже если это очевидно, нельзя недооценивать важность данного процесса.