

*Амиров Азамат Жанбулатович, доктор Ph.D*

*Факультет инновационных технологий*

*Карагандинский государственный технический университет, г. Караганда*

*Динмухаммедулы Диас, магистр, преподаватель*

*Факультет инновационных технологий*

*Карагандинский государственный технический университет, г. Караганда*

## **ОЦЕНКА ВРЕМЕНИ ДЛЯ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**Аннотация:** В статье рассматривается одна из важных этапов создания программного обеспечение, тестирование. Проводится анализ каждого этапа и только после этого программное обеспечение может быть передано на реализацию и дальнейшее техническое сопровождение. Для большего понимания данной статьи была найдена золотая середина между фундаментальной теорией и необходимой практикой, а также добавлены примеры решения и контрольные вопросы.

**Ключевые слова:** время, тестирование, задание, среда, интерфейс, приложение.

**Abstract:** the article discusses one of the important stages of software development, testing. Each stage is analyzed and only after that the software can be transferred for implementation and further technical support. For a better understanding of this article, a Golden mean was found between the fundamental theory and the necessary practice, as well as added examples of solutions and control questions.

**Keywords:** time, testing, task, environment, interface, application.

## **Введение**

В процессе создания успешного программного продукта возникает неизбежная проблема поиска баланса между качеством и датой выпуска программного продукта. Тестирование позволяет получить продукт, который удовлетворяет всем требованиям. Но покрытие каждого продукта сопряжено с различными тестовыми примерами, и их компиляция занимает слишком много времени. Правильно подготовленный процесс тестирования должен обеспечивать требуемый уровень качества без превышения времени и бюджета проекта. Если время тестирования было оценено неверно, это может привести либо к задержке поставки продукта, либо к снижению его качества и конкурентоспособности. Оценка тестирования программного обеспечения является довольно сложным и объемным процессом, но его значение для создания успешного проекта не следует недооценивать.

Эта статья содержит рекомендации о том, как проводить оценку тестирования программного обеспечения, что, как мы надеемся, поможет вам получить более реалистичные и функциональные оценки времени для нового проекта.

## **Разложение тестовых заданий**

Согласно статистике оценки QA, тестирование однокомпонентного консольного приложения занимает около 20% времени его разработки. Тестирование двухкомпонентного консольного приложения занимает 20-30% времени его разработки, приложения с графическим интерфейсом - 30-35%, распределенного приложения с графическим интерфейсом - 35-50%. Но каждый проект и каждая команда уникальны, поэтому эта временная оценка довольно грубая и не включает в себя некоторые риски.

Чтобы сделать оценку времени тестирования более точной и реалистичной, вы должны использовать метод декомпозиции, то есть вы должны разделить процесс тестирования на несколько частей и оценить время для каждой из них.

Как правило, процесс тестирования нового продукта можно разделить на 6 основных этапов:

1. Процесс планирования.
2. План тестирования и разработка тестового примера.
3. Тестирование конфигурации среды.
4. Выполнение тестовых случаев.
5. Отладка тестовых случаев после первого запуска или после смены продукта.
6. Регрессионное тестирование [2; 3; 4].

### **Оценка планирования процесса тестирования**

Этот этап, в свою очередь, состоит из двух подэтапов: исследование проекта и разработка стратегии тестирования.

Исследование проекта может проводиться по-разному в разных компаниях и командах: чтение и анализ проектной документации, первоначальная встреча, обсуждение деталей проекта с менеджерами и т. Д. В среднем мы рекомендуем потратить один-два дня на эту задачу, не обязательно, чтобы это задание занимало целый рабочий день, но оценка 1-2 дня позволит вам иметь достаточно времени в случае появления новых идей и вопросов.

*Дополнительное время требуется в случаях:*

- Проект достаточно масштабный, имеет много документации.
- Работа с такого рода проектом выполняется впервые, и для исследования требуется дополнительное время.
- Специалисты, которые принимают участие в обсуждении проекта, имеют разные графики работы и географически разбросаны. В таких случаях требуется дополнительное время для организации встреч.

Разработка стратегии тестирования предполагает определение ряда важных нюансов будущей работы:

- Какие виды тестирования будут проводиться.
- Какие специалисты требуются для разработки планов тестирования и тестовых случаев.

- Сколько специалистов проведут тестирование проекта и какими навыками они должны обладать.

В нашей практике для определения необходимых видов и типов и эффективных методов тестирования программного обеспечения мы опираемся на готовый шаблон, который включает все виды тестирования, которые могут быть актуальными.

В зависимости от типа проекта, его требований к качеству и других факторов, модель корректируется. Такая подстройка занимает около часа.

Внимание: на данном этапе определение специалистов для работы над проектом поможет более точно оценить время, так как мы можем учесть их навыки и опыт. Это также поможет включить в график тестирования свои каникулы и выходные [1].

### **План тестирования и оценка тестового набора, и разработка**

Разработка плана тестирования и тестовых случаев является довольно трудоемким процессом и требует значительных временных затрат. При планировании следует исходить из стратегии тестирования, разработанной на предыдущем этапе. Стратегия тестирования показывает, какие типы тестов должны быть созданы, а также помогает установить их приоритет.

В нашей практике мы руководствуемся правилом, согласно которому одно требование к продукту должно быть покрыто как минимум пятью контрольными случаями. С помощью этого правила мы можем приблизительно рассчитать количество необходимых тестовых случаев и время их создания.

Время разработки тестового примера зависит от сложности плана тестирования, но в среднем разработка одного тестового примера занимает 10 минут. В общем случае разработка плана тестирования без тестовых случаев и его проверка могут занять два-три дня. Соответственно, если проект требует контрольных примеров, вам следует оценить дополнительное время для их разработки.

*Некоторые особенности планирования этого этапа:*

- Если специалист готовит план тестирования и тестовые случаи впервые, вам следует оценить больше времени, чем для более опытного специалиста.

- Если в проекте используются новые для команды технологии, вам следует учитывать время для его исследования. В зависимости от сложности технологии, а также от квалификации специалистов, которые будут выполнять эту задачу, дополнительное время для исследования может составлять от одного дня до нескольких недель. Вы не должны пренебрегать этим ни в коем случае. Если план тестирования подготовлен без учета специфики проекта, это в лучшем случае приведет к увеличению времени для внесения в него изменений на стадии тестирования, а в худшем - к снижению качества продукта.

- В зависимости от типов тестирования и проекта вам может потребоваться время для создания тестовых данных. Это также следует учитывать на этом этапе.

- Порядок разработки тестовых случаев должен соответствовать порядку их запуска. Для приоритета теста может быть два варианта:

- Во-первых, вы должны запустить тесты, которые охватывают первоочередные модули и функции проекта.

- Во-вторых, вы должны запустить тесты для модулей проекта, готовых к тестированию.

В зависимости от порядка выполнения теста, вы должны подготовить порядок создания плана тестирования и тестовых случаев. Это также может повлиять на окончательную оценку.

- Если у компании есть опыт тестирования подобных проектов, то использование старых планов тестирования и тестовых случаев в качестве основы сократит предполагаемое время.

- Время для проверки подготовленных планов тестирования и контрольных примеров должно быть выделено как для вновь созданных, так и для измененных старых. При планировании этого времени вы должны учитывать рабочую нагрузку и личные графики специалистов, которые будут выполнять проверку.

- Чем опытнее специалист, создающий план тестирования и контрольные примеры, тем меньше времени требуется для проверки и дальнейших изменений [2].

### **Конфигурация тестовой среды**

Время, необходимое для этого этапа работы, зависит от следующих факторов:

- Есть ли необходимость покупать какое-то оборудование или нет? Когда необходимо расширить испытательную лабораторию, следует планировать время с учетом специфики и возможностей компании. Если оборудование продается, и компания может его купить сразу, время будет значительно сокращено по сравнению со случаем, когда конкретное оборудование должно быть поставлено иностранным заказчиком или, когда компания не может себе это позволить в данный момент.

- Время на установку и настройку тестовой среды зависит от квалификации специалиста и опыта работы с аналогичными тестовыми средами. Если в компании есть специальный отдел для таких задач, потребуется меньше времени по сравнению со случаем, когда специалисты по QA будут выполнять эту работу.

Время на установку и настройку тестовой среды напрямую зависит от сложности последней. Обычно настройка среды тестирования для проекта среднего размера, который работает с популярными операционными системами и не требует сложных системных решений, занимает от одного часа до одного дня. В другом случае требуется дополнительная оценка времени в зависимости от особенностей задачи.

### **Выполнение тестовых случаев**

В нашей практике мы используем правило, согласно которому выполнение одного контрольного примера занимает у специалиста по обеспечению качества около 5 минут. План тестирования содержит тесты различной сложности и масштаба: некоторые тесты могут быть выполнены за 1 минуту, другие - за 10 минут. В результате средняя продолжительность теста составляет 5 минут.

Рекомендуется увеличить время одного теста до 10 минут, если тестирование выполняется младшим специалистом по обеспечению качества. Таким образом, мы можем принять во внимание риски, которые существуют, если нет необходимого опыта работы.

Одна из основных трудностей планирования этого этапа работы заключается в том, что вы не можете точно предсказать количество ошибок, обнаруженных при тестировании, а также сложность их воспроизведения. В среднем на написание отчета об одной ошибке уходит 10-15 минут. Чем больше ошибок найдено, тем больше времени требуется для отчетов по каждому из них. Если ошибка слишком сложна, даже нескольких часов недостаточно, чтобы узнать ее точное местоположение. Существуют методы, позволяющие оценить количество возможных ошибок в каждой последующей версии продукта. Трудно точно определить количество ошибок в новой версии.

Чтобы тестирование не превышало запланированное время, вы должны включить в график время для различных рисков. Дополнительное время на подготовку ряда отчетов о найденных ошибках, а также время нахождения наиболее сложных ошибок будет включено в общие риски тестирования. В соответствии с нашими методами оценки тестирования, мы рекомендуем вам добавить около 20-25% времени для таких случаев к вашей окончательной оценке.

В любом случае, вы должны следовать своему здравому смыслу. Если тестирование версии продукта оценивается в 10 часов, а локализация одной найденной ошибки занимает более 2-3 часов, логично отложить ее до конца фазы тестирования (если ошибка не слишком критична или блокируется). После окончания фазы тестирования, если осталось время, вы можете вернуться к работе над этой ошибкой. В этом случае вы обеспечите наличие всех необходимых результатов тестирования, не выбрасывая проект из графика [2; 5].

**Отладка тестовых случаев после первого запуска или изменения продукта**

Этот этап работы занимает около 10-15% времени, необходимого для создания плана тестирования и тестовых случаев.

Также требуемое время напрямую зависит от времени, отведенного на рассмотрение плана тестирования и тестовых случаев сразу после их подготовки. Если проверка была достаточно тщательной, время для дальнейшей переделки после прогона может уменьшиться. Но время для рецензирования должно учитываться адекватно [8].

### **Регрессионное тестирование**

При проведении оценки регрессионного тестирования для новых версий продукта необходимо учитывать следующие нюансы:

- Какой тип тестирования требуется на этом этапе? Необходимо ли снова проводить полное тестирование или просто тестирование дыма для промежуточной версии продукта?
- Количество тестовых примеров в плане тестирования может быть изменено после его корректировки.

Когда тестирование выполняется одним и тем же специалистом, время проведения тестов может уменьшиться, так как тесты не будут для него новыми.

Если проект поддерживает несколько операционных систем, необходимо протестировать его работу на всех операционных системах, требуемых спецификацией. Когда продукт имеет клиент-серверную архитектуру, также важно запускать тесты на разных комбинациях систем. Часто невозможно проверить все комбинации, потому что требуются большие временные ресурсы, и они не сопоставимы с общим временем разработки.

В таких случаях важно максимально охватить матрицу комбинаций систем. Например, если приложение поддерживает пять разных систем на стороне сервера и пять систем на стороне клиента, вы должны выбрать комбинации, которые охватывают все эти системы без повторов. Это означает, что вам следует избегать варианта, когда клиент XP SP3 тестируется со всеми возможными системами на стороне сервера или наоборот. Вы просто пытаетесь охватить каждую систему в тестовых комбинациях.



Если версия продукта является промежуточной, полное тестирование должно проводиться на нескольких системах, которые наиболее критичны для пользователей. Проверка дыма должна проводиться на остальных системах. Принимая во внимание системные приоритеты для вашего продукта, вы можете заполнить матрицу максимально эффективно и провести тестирование за ограниченный период времени.

Когда готовится матрица комбинаций систем, актуальная для данного этапа разработки, легко оценить общее время тестирования версии, используя такую методику оценки:

$$T = (T(A) * K(A) + T(F) * K(F) + T(S) * K(S)) * 1,25,$$

где

T () - время прохождения плана тестирования на одной конфигурации.

K () - количество конфигураций для прохождения.

A - приемочные испытания;

S - тестирование дыма.

F - полное тестирование [6; 7].

### **Заключение**

Трудно указать точные методы оценки в тестировании программного обеспечения, так как QA - сложный процесс с высоким риском, и во всех его оценках всегда есть некоторое отклонение. Вот почему эффективно комбинировать различные методы и методы оценки тестирования программного обеспечения с учетом специфики проекта и команды тестирования с пониманием факторов, влияющих на затраты, время и ресурсы, таких как знание команды или модель разработки конкретного гибкого проекта. ,

Метод, который мы предлагаем вам, поможет вам составить базовую оценку для тестирования вашего продукта; учет всех его особенностей поможет повысить точность оценки и сделать ее максимально реалистичной.

Итак, общая формула для оценки тестирования следующая (наш шаблон оценки):

T = T (изучение спецификаций + стратегия тестирования) + T (подготовка документации к тестированию) + T (подготовка среды тестирования) + T (первый запуск + обновление документации по тестированию) + T (регрессионное тестирование).

### **Библиографический список:**

1. Sam Kaner. Software testing, 2000, 544с.
2. Myers G. The Art of Testing Programs. Finance and Statistics, 1982, 176 с.
3. Myers G. Software Reliability, 1980, 360 с.
4. Van Tassel. Style, development, efficiency, debugging and testing programs, 1981, 322 с.
5. Goodman S., Hudetniemi S. Introduction to the development and analysis of algorithms, 1981, 371 с.
6. Ziegler K. Methods of designing software systems, 1985, 328 с.
7. Brooks F.P. How software systems are designed and created, 1979, 228 с.
8. Fox J. Software and its development, 1982, 299 с.