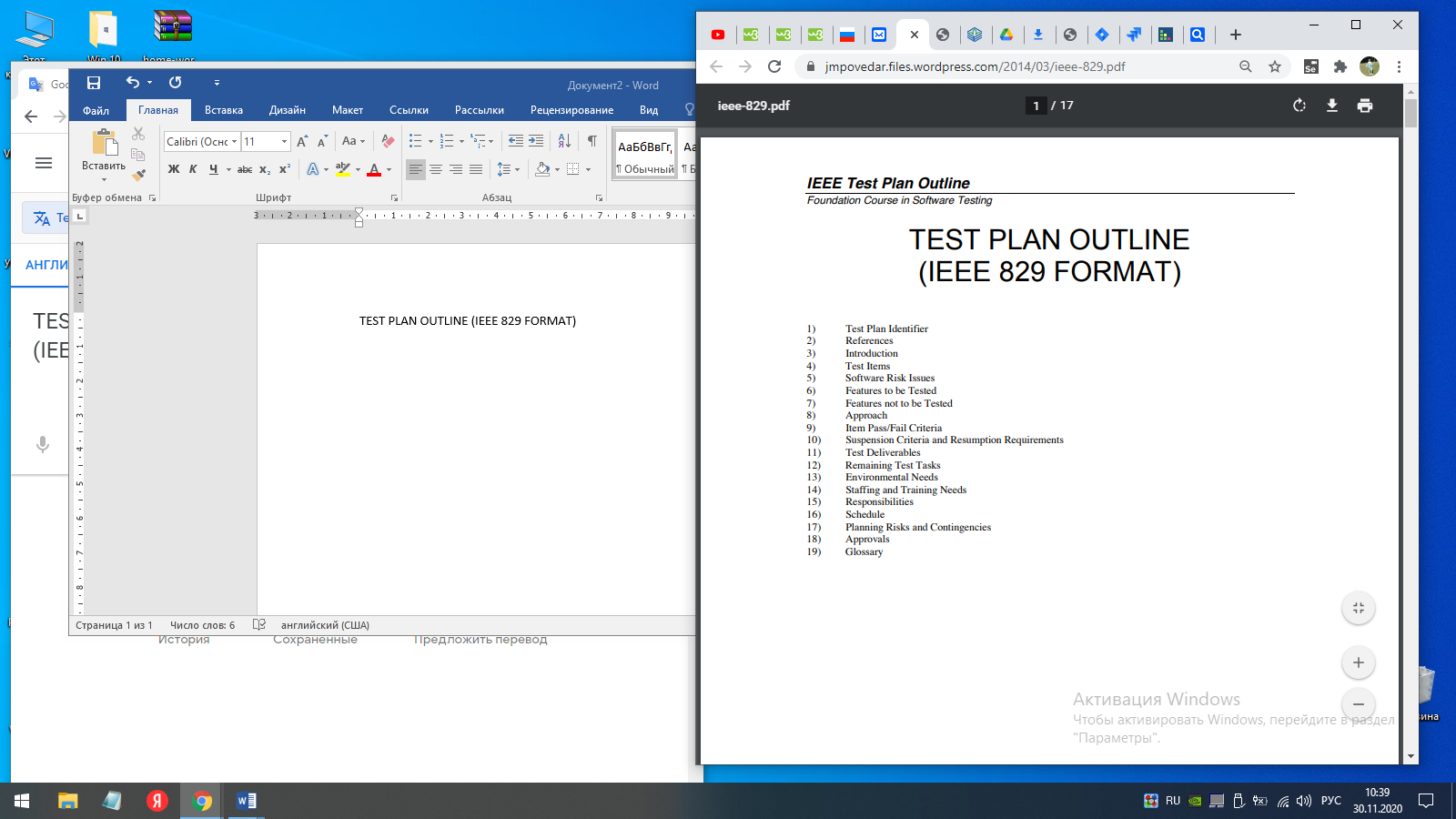
TEST PLAN OUTLINE (IEEE 829 FORMAT)



ШАБЛОН ПЛАНА ТЕСТИРОВАНИЯ IEEE

1 ИДЕНТИФИКАТОР ПЛАНА ИСПЫТАНИЙ

Некий тип уникального номера компании, сгенерированного для идентификации этого плана тестирования, его уровня и

уровень программного обеспечения, к которому он относится. Желательно, чтобы уровень плана тестирования был таким же, как и

уровень соответствующего программного обеспечения. Число может также определить, является ли план тестирования Генеральным планом,

План уровня, план интеграции или любой другой уровень плана, который он представляет. Это должно помочь в

согласование версий ПО и тестового ПО в рамках управления конфигурацией.

Имейте в виду, что планы тестирования похожи на другую документацию по программному обеспечению, они динамичны по своей природе.

и должны обновляться. Следовательно, у них будут номера ревизий.

Вы можете указать автора и контактную информацию, включая историю изменений.

информация как часть раздела идентификаторов или как часть введения.

2 ССЫЛКИ

Перечислите все документы, поддерживающие этот план тестирования. См. Фактический номер версии / выпуска

документ, хранящийся в системе управления конфигурацией. Не дублируйте текст

из других документов, так как это снизит жизнеспособность этого документа и увеличит

усилия по обслуживанию. Документы, на которые можно ссылаться, включают:

!"План проэкта

! »Технические требования

! "Проектный документ высокого уровня

! »Детальный конструкторский документ

! »Стандарты процессов разработки и тестирования

! »Методические указания и примеры

! »Корпоративные стандарты и правила

3 ВВЕДЕНИЕ

Укажите цель плана, возможно, указав уровень плана (мастер и т. Д.). Это

по сути, это сводная часть плана.

Вы можете включить любые ссылки на другие планы, документы или предметы, содержащие

информация, относящаяся к этому проекту / процессу. При желании вы можете создать раздел ссылок

содержать все справочные документы.

Определите Объем плана по отношению к плану Программного проекта, к которому он относится. Другие

элементы могут включать, ресурсные и бюджетные ограничения, объем работ по тестированию, способы тестирования

относится к другой деятельности по оценке (анализ и обзоры), и, возможно, процесс

используется для управления изменениями, коммуникации и координации ключевых действий.

Поскольку это «Резюме», дайте информацию кратко и по существу.

4 ТЕСТОВЫХ ПУНКТА (ФУНКЦИИ)

Это то, что вы собираетесь протестировать в рамках этого плана тестирования. По сути, что-то

вы будете тестировать, список того, что нужно проверить. Это может быть разработано из программного обеспечения

описи приложений, а также другие источники документации и информации.

Это может контролироваться и определяться вашим локальным процессом управления конфигурацией (CM), если

у тебя есть. Эта информация включает номера версий, требования к конфигурации, где

необходимо (особенно, если поддерживается несколько версий продукта). Он также может включать ключ

проблемы с графиком доставки критических элементов.

Помните: то, что вы тестируете, - это то, что вы собираетесь доставить Клиенту.

Этот раздел можно сориентировать на уровень плана тестирования. Для более высоких уровней это может быть

приложение или функциональная область, для более низких уровней это может быть программа, модуль, модуль или сборка.

5 ВОПРОСЫ РИСКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Определите, какое программное обеспечение следует тестировать и каковы критические области, например:

A. Доставка стороннего продукта.

Б. Новая версия интерфейсного программного обеспечения

C. Способность использовать и понимать новый пакет / инструмент и т. Д.

D. Чрезвычайно сложные функции

E. Модификации компонентов с историей отказов

F. Плохо документированные модули или запросы на изменение

Программному обеспечению присущи некоторые риски, такие как сложность; их необходимо идентифицировать.

A. Безопасность

Б. Несколько интерфейсов

В. Влияние на клиента

D. Правительственные постановления и правила

Еще одна ключевая область риска - неправильное понимание исходных требований. Это может произойти в

уровни управления, пользователя и разработчика. Помните о расплывчатых или неясных требованиях и

требования, которые невозможно проверить.

Прошлая история дефектов (ошибок), обнаруженных во время модульного тестирования, поможет выявить потенциальные

области программного обеспечения, которые являются рискованными. Если модульное тестирование обнаружило большое количество

дефекты или склонность к дефектам в определенной области программного обеспечения, это показатель

потенциальных будущих проблем. По своей природе дефекты группируются и сливаются вместе. Если оно

был дефектным ранее, он, скорее всего, по-прежнему будет подвержен дефектам.

Один хороший способ определить, где находятся риски, - это провести несколько мозговых штурмов.

! "Начните с идей, например, что меня беспокоит в этом проекте / приложении.

6 ХАРАКТЕРИСТИК ДЛЯ ПРОВЕРКИ

Это список того, что должно быть протестировано с точки зрения ПОЛЬЗОВАТЕЛЕЙ того, что делает система.

Это не техническое описание программного обеспечения, а представление функций ПОЛЬЗОВАТЕЛЯМИ.

Установите уровень риска для каждой функции. Используйте простую шкалу оценок, например (H, M, L): High,

Средний и низкий. Эти типы уровней понятны Пользователю. Ты должен быть

готовы обсудить, почему был выбран тот или иной уровень.

Следует отметить, что Раздел 4 и Раздел 6 очень похожи. Единственная настоящая разница - это

точка зрения. Раздел 4 представляет собой техническое описание типа, включая номера версий и другие

техническая информация, а раздел 6 - с точки зрения пользователя. Пользователи не понимают терминология технического программного обеспечения; они понимают функции и процессы, связанные с их

рабочие места.

7 ФУНКЦИЙ, НЕ ПОДЛЕЖАЩИЕ ТЕСТИРОВАНИЮ

Это список того, что НЕ подлежит тестированию с точки зрения пользователей того, что система

делает и представление управления конфигурацией / контроля версий. Это не технический

описание программного обеспечения, но представление функций ПОЛЬЗОВАТЕЛЯМ.

Определите, ПОЧЕМУ функция не должна тестироваться, может быть любое количество причин.

! »Не включать в эту версию Программного обеспечения.

! "Низкий риск, уже использовался и считается стабильным.

! "Будет выпущен, но не протестирован и не задокументирован как функциональная часть выпуска этого

версия программного обеспечения.

Разделы 6 и 7 напрямую связаны с Разделами 5 и 17. Что будет и что не будет проверяться, так это

напрямую зависит от уровней допустимого риска в рамках проекта, а что не получается

тестирование влияет на уровень риска проекта.

8 ПОДХОД (СТРАТЕГИЯ)

Это ваша общая стратегия тестирования для этого плана тестирования; он должен соответствовать уровню

план (основной, принятие и т. д.) и должен согласовываться со всеми высшими и низшими уровнями

планы. Следует определить общие правила и процессы.

! "Можно ли использовать какие-либо специальные инструменты и какие они?

! »Потребует ли инструмент специальной подготовки?

! »Какие метрики будут собираться?

! »На каком уровне будет собираться каждая метрика?

! "Как управлять конфигурацией?

! »Сколько различных конфигураций будет протестировано?

! "Оборудование

!"Программного обеспечения

! »Комбинации аппаратного, программного обеспечения и пакетов других поставщиков

! »Какие уровни регрессионного тестирования будут проводиться и сколько на каждом уровне тестирования?

! »Будет ли регрессионное тестирование основываться на серьезности обнаруженных дефектов?

! "Как элементы требований и дизайна, которые не имеют смысла или

непроверяемый будет обработан?

Если это основной план тестирования, то общий подход к тестированию проекта и требования к покрытию

также должны быть идентифицированы.

Уточните, есть ли особые требования к тестированию.

! »Будет протестирован только полный компонент.

! »Определенный сегмент группы функций / компонентов должен быть протестирован вместе.

Другая информация, которая может быть полезна при настройке подхода:

! "Среднее время наработки на отказ, среднее время наработки на отказ - если это действительное измерение для данного теста

и если данные доступны.

! »SRE, Software Reliability Engineering - если эта методология используется и

информация имеется.

Как будут проводиться встречи и другие организационные процессы?

9 КРИТЕРИИ ПРОЙДЕННОСТИ / НЕИСПРАВНОСТИ ЭЛЕМЕНТА

Каковы критерии завершения для этого плана? Это важный аспект любого плана тестирования и

должен соответствовать уровню плана.

! "На уровне модульного теста это могут быть такие элементы, как:

! "Все тестовые примеры выполнены.

! "Указанный процент дел, завершенных с процентом, содержащим некоторое число

мелких дефектов.

! "Инструмент покрытия кода указывает на весь охватываемый код.

! "На уровне генерального плана тестирования это могут быть такие элементы, как:

! "Все планы нижнего уровня выполнены.

! »Указанное количество планов, выполненных без ошибок, и процент с незначительными

дефекты.

Это может быть критерий уровня отдельного тестового примера, план уровня единицы или общий

функциональные требования для планов более высокого уровня.

Какое количество и степень серьезности обнаруженных дефектов?

! »Можно ли сравнить это с общим количеством дефектов? Это может быть невозможно, так как

некоторые дефекты никогда не обнаруживаются.

! "Дефект - это то, что может вызвать сбой, и его можно оставить в

применение.

! "Отказ является результатом дефекта, видимого Пользователем, сбоя системы и т. Д.

10 КРИТЕРИИ ПРИОСТАНОВКИ И ТРЕБОВАНИЯ К ВОЗОБНОВЛЕНИЮ

Знайте, когда нужно сделать паузу в серии тестов.

! "Если количество или тип дефектов достигает точки, при которой последующее тестирование не имеет значения,

нет смысла продолжать тест; вы просто тратите ресурсы.

Укажите, что представляет собой остановку теста или серии тестов и каков приемлемый уровень

дефектов, которые позволят продолжить тестирование после дефектов.

Тестирование после действительно фатальной ошибки приведет к возникновению условий, которые могут быть определены как дефекты, но

фактически являются ошибочными ошибками, вызванными более ранними дефектами, которые игнорировались.

11 ИСПЫТАТЕЛЬНЫЕ МАТЕРИАЛЫ

Что должно быть выполнено в рамках этого плана?

! "Документ плана испытаний.

! "Тестовые кейсы.

! »Технические условия на дизайн.

! »Инструменты и их выходы.

! »Симуляторы.

! »Статические и динамические генераторы.

! "Журналы ошибок и журналы выполнения.

! »Отчеты о проблемах и корректирующие действия.

Одна вещь, которая не является результатом тестирования, - это само программное обеспечение, которое указано в элементах тестирования и

доставляется путем разработки.

12 ОСТАВШИХСЯ ТЕСТОВЫХ ЗАДАНИЙ

Если это многоэтапный процесс или если приложение будет выпускаться поэтапно, может

быть частями приложения, которые не рассматриваются в этом плане. Эти области необходимо определить, чтобы

избежать путаницы, если в будущих функциях будет сообщено о дефектах. Это также

позволяют пользователям и тестировщикам избежать неполных функций и предотвратить трату ресурсов

погоня за недочетами.

Если проект разрабатывается как многосторонний процесс, этот план может охватывать только часть

общих функций / возможностей. Этот статус необходимо определить, чтобы в других областях

планы, разработанные для них, и избежать бесполезной траты ресурсов на отслеживание дефектов, не связанных с

этот план.

Когда третья сторона разрабатывает программное обеспечение, этот раздел может содержать описания тех

тестовые задания, принадлежащие как внутренним, так и внешним группам.

13 ЭКОЛОГИЧЕСКИЕ ПОТРЕБНОСТИ

Существуют ли какие-либо особые требования к этому плану тестирования, например:

! »Специальное оборудование, такое как симуляторы, статические генераторы и т. Д.

! "Как будут предоставляться тестовые данные. Существуют ли особые требования к сбору или особые

диапазоны данных, которые необходимо предоставить?

! »Сколько тестов будет выполнено для каждого компонента функции, состоящей из нескольких частей?

! »Особые требования к питанию.

! »Определенные версии другого поддерживающего программного обеспечения.

! "Ограниченное использование системы во время тестирования.

14 ПОТРЕБНОСТИ В ПЕРСОНАЛЕ И ОБУЧЕНИИ

Обучение работе с приложением / системой.

Обучение работе с любыми инструментами тестирования.

Раздел 4 и Раздел 15 также влияют на этот раздел. Что проверять и кто за это отвечает

для тестирования и обучения.

15 ОБЯЗАННОСТИ

Кто отвечает?

Этот выпуск включает в себя все области плана. Вот некоторые примеры:

! »Установка рисков.

! "Выбор функций для тестирования, а не для тестирования.

! »Установка общей стратегии для этого уровня плана.

! »Обеспечение наличия всех необходимых элементов для тестирования.

! "Обеспечение разрешения конфликтов расписания, особенно, если тестирование проводится на

система производства.

! »Кто проводит необходимое обучение?

! »Кто принимает критические решения« годен / не годен »для элементов, не включенных в планы тестирования?

16 РАСПИСАНИЕ

Должен быть основан на реалистичных и подтвержденных оценках. Если сметы на развитие

заявка неточна, весь план проекта выскользнет, ​​а тестирование является частью

общий план проекта.

! "Как мы все знаем, первая часть плана проекта, которую нужно сократить, когда дело доходит до критического времени в

конец проекта - это тестирование. Обычно все сводится к решению: «Давайте положим

что-то выходит, даже если это не совсем так хорошо ». И, как мы все знаем, это

обычно худшее возможное решение.

Здесь также следует рассмотреть вопрос о том, как будет устраняться отставание в графике.

! »Если пользователи заранее знают, что пробуксовка в разработке вызовет пробуксовку

испытания и общая поставка системы, они могут быть немного более терпимыми, если

они знают, что в их интересах получить лучше протестированное приложение.

! »Излагая здесь эффекты, у вас есть возможность обсудить их до их

фактическое появление. Вы даже можете заставить пользователей заранее согласиться с некоторыми дефектами, если

расписания листов.

На этом этапе следует определить все важные вехи с их отношением к

процесс разработки идентифицирован. Это также поможет в выявлении и отслеживании потенциальных

проскальзывание графика из-за процесса тестирования.

Всегда лучше связать все даты тестирования напрямую с соответствующими датами разработки. Этот

предотвращает восприятие тестовой команды как причины задержки. Например, если система

тестирование должно начаться после доставки финальной сборки, затем тестирование системы начинается на следующий день после

Доставка. Если доставка задерживается, тестирование системы начинается со дня доставки, а не в

Конкретная дата. Это называется зависимым или относительным свиданием.

17 ПЛАНИРОВАНИЕ РИСКОВ И УСЛОВНЫХ ОБЯЗАТЕЛЬСТВ

Каковы общие риски для проекта с упором на процесс тестирования?

! »Отсутствие кадровых ресурсов при начале тестирования.

! »Отсутствие необходимого оборудования, программного обеспечения, данных или инструментов.

! »Несвоевременная поставка программного обеспечения, оборудования или инструментов.

! »Задержки в обучении работе с приложением и / или инструментами.

! »Изменения исходных требований или дизайна.

Укажите, что будет сделано для различных мероприятий, например:

! »Определение требований будет завершено к 1 января 19XX г., и, если требования

изменить после этой даты, будут предприняты следующие действия.

! "График тестирования и график разработки переместятся на соответствующее количество

дней. Это случается редко, поскольку у большинства проектов, как правило, есть фиксированные сроки сдачи.

! »Количество выполняемых тестов будет уменьшено.

! »Количество допустимых дефектов будет увеличено.

! »Эти два элемента могут снизить общее качество поставляемого продукта.

! "Ресурсы будут добавлены к команде тестирования.

! "Группа тестирования будет работать сверхурочно.

! "Это может повлиять на боевой дух команды.

! »Объем плана может быть изменен.

! "Возможна некоторая оптимизация ресурсов. Этого следует избегать, если возможно,

по понятным причинам.

! "Можно было просто ВЫЙТИ. Довольно крайний вариант, мягко говоря.

Руководство обычно неохотно принимает сценарии, подобные приведенному выше, даже если они

видели это в прошлом.

Важно помнить, что, если вы вообще ничего не делаете, обычный результат

тестирование сокращено или полностью исключено, ни то, ни другое не должно быть приемлемым вариантом.

18 РАЗРЕШЕНИЙ

Кто может утвердить процесс как завершенный и позволить проекту перейти на следующий уровень

(в зависимости от уровня плана)?

На уровне генерального плана тестирования это могут быть все участвующие стороны.

При определении процесса утверждения помните, кто является аудиторией.

! "Аудитория плана уровня модульного тестирования отличается от аудитории интеграции, системы или

план мастер-уровня.

! »Уровни и типы знаний на разных уровнях также будут разными.

! "Программисты очень техничны, но могут не иметь четкого представления об общих

бизнес-процесс, управляющий проектом.

! "Пользователи могут иметь разный уровень деловой хватки и очень слабые технические навыки.

! »Всегда остерегайтесь пользователей, которые заявляют о высоком уровне технических навыков, и программистов, которые

претендовать на полное понимание бизнес-процесса. Эти типы людей могут вызвать больше

вред, чем польза, если у них нет навыков, которыми, по их мнению, обладают.

19 ГЛОССАРИЙ

Используется для определения терминов и сокращений, используемых в документе, и тестирования в целом, чтобы исключить

путаница и способствовать последовательному общению.