Notepad

Release 10586.71

Test plan for Notepad

Version 0.1

Draft

|  |  |
| --- | --- |
| Project team: | Vera Getman  Ganna Kaplun  Artem Zagarbul |
|  |  |
|  |  |
|  |  |

September 2017

# Test Plan Identifier

Тест план должен иметь уникальное «короткое» имя, дату версии, номер версии процедуры, имя автора версии и контактная информация, лист регистраций и изменений.

# Introduction

Целью составления данного тест плана является описание процесса тестирования приложения Notepad, а по итогу прохождение всех тест-кейсов успешно. Документ позволяет получить представление о действиях по тестированию программы. Документ предназначен группе тестирования для ознакомления с перечнем предстоящих работ, анализа и разбиения на подзадачи для всех людей в команде.

# Test Items

Notepad (блокнот) — простой текстовый редактор, являющийся частью операционных систем Microsoft Windows, предназначенный для создания и изменения текстовых данных. Кроме Windows, Блокнот способен выполняться также в [ReactOS](https://ru.wikipedia.org/wiki/ReactOS" \o "ReactOS) и [Wine](https://ru.wikipedia.org/wiki/Wine" \o "Wine). По умолчанию, Блокнот создает, открывает и сохраняет документы с расширением txt. Позволяет работать с текстами в кодировках ANSI и Unicode, а также выполнять преобразование из одного формата в другой.

# Features To Be Tested

Ниже представлен список необходимых функций для тестирования:

1. Создание файла
2. Сохранение файла
3. Открытие файла
4. Редактирование файлов
5. Поиск в тексте
6. Печать документа
7. Совместимость с операционными системами [ReactOS](https://ru.wikipedia.org/wiki/ReactOS) и [Wine](https://ru.wikipedia.org/wiki/Wine" \o "Wine)

I. Копировать, вставить

J. Работа приложения в других версиях Windows (7,8,10,XP)

# Features Not To Be Tested

Ниже представлен список функций которые не будут рассмотрены в данном тест плане, вследствие недостатка ресурсов:

1. Установку и удаление приложения
2. Тестирование производительности
3. Надежности приложения
4. Тестирование стабильности
5. Тестирование безопасности

# Approach

В процессе тестирования приложения Notepad будет применено функциональное тестирование по документации. Планируется три этапа проведения процесса тестирования. Первый этап заключается в составлении чек-листа необходимых для проверки функций.

На втором этапе будет написание тест-кейсов.

На третьем этапе, в ходе проведения функционального тестирования, планируется проведение smoke testing и sanity testing основанное на корректном сценарии работы пользователя с приложением. Так же будет проведено ряд негативных тестов.

Тестирование предполагается вести в ручном режиме, без использования автоматизированных систем. Для сокращения количества тестов будет использоваться техника разбиения на классы эквивалентности.

# Item Pass/Fail Criteria

Процесс тестирования считается завершенным при следующих условиях выполнения:

1. Все тесты пройдены
2. Указан процент случаев содержащих определенное количество дефектов
3. Тестовое покрытие проверено и является достаточным

# Suspension Criteria And Resumption Requirements

1. Количество дефектов достигает точки, где последующее тестирование не имеет значения
2. Фатальная ошибка при запуске приложения
3. Отсутствие среды для тестирования
4. Тестирования приостанавливается, пока причина при которой вызван сбой не будет решена

# Test Deliverables

1. Тест план
2. Чек-лист
3. Тест-кейсы
4. Отчеты об ошибках
5. Cпецификации тестовых случаев

# Environmental Needs

Список операционных систем для проверки открытия файлов созданных в Notepad:

1. Windows 10
2. Windows 8
3. Windows 7
4. Windows XP
5. [ReactOS](https://ru.wikipedia.org/wiki/ReactOS)
6. [Wine](https://ru.wikipedia.org/wiki/Wine)

# Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
|  | Kaplun | Getman | Zagarbul |
| Тест план | Х |  |  |
| Чек-лист |  | Х | Х |
| Тест-кейсы |  | Х | Х |
| Ответственность за критические решения | Х |  |  |

# Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone Task | Effort | Start Date | End Date |
| Тест план | 1 чел/час | 01.10 | 03.10 |
| Чек-лист | 2 чел/час | 03.10 | 05.10 |
| Тест-кейсы | 2 чел/час | 03.10 | 05.10 |

# Approvals

|  |  |
| --- | --- |
| Team Leader - Artem Bokiy |  |
| Test Manager - Olga Maklakova |  |
| Test Manager - Sergey Zelinsky |  |

**Идентификатор плана тестирования – Test plan Identifier**

Некоторый тип уникальной компании сгенерировал число, чтобы определить этот план тестирования, его уровень и

уровень программного обеспечения, к которому он относится. Предпочтительно уровень плана тестирования будет таким же, как и

связанных с программным обеспечением. Номер может также определить, является ли план тестирования генеральным планом,

План уровня, план интеграции или любой уровень плана, который он представляет. Это поможет

координируя программное обеспечение и версии тестовых программ в рамках управления конфигурацией.

• Уникальное «короткое» имя для плана тестирования

• Дата версии и номер версии процедуры

• Автор версии и контактная информация

• Лист регистраций изменений

Имейте в виду, что планы тестирования подобны другим документам программного обеспечения, они динамичны в

природы и должны быть обновлены. Поэтому они будут иметь номера ревизий.

Вы можете включить автора и контактную информацию, включая историю изменений

информацию как часть раздела идентификатора как часть введения.

**Введение - Introduction**

Укажите цель Плана, возможно, определив уровень плана (мастер и т. Д.). Эта

является, по сути, исполнительной частью плана.

Вы можете включить любые ссылки на другие планы, документы или элементы, которые содержат

информация, относящаяся к этому проекту / процессу. Если предпочтительнее, вы можете создать раздел ссылок

чтобы содержать все справочные документы.

• Авторизация проекта

• План проэкта

• План обеспечения качества

• План управления конфигурацией

• Соответствующие политики и стандарты

• Для планов более низкого уровня, укажите план (ы) более высокого уровня

Определите сферу действия плана в отношении плана Программного проекта, к которому он относится.

Другие элементы могут включать ограничения ресурсов и бюджета, объем усилий по тестированию, как

тестирование относится к другим мероприятиям по оценке (анализ и обзоры), и возможно

использоваться для контроля изменений и связи и координации ключевых видов деятельности.

Поскольку это «Краткое резюме», держите информацию кратким и точным.

**Тестовые задания – Test Items**

Это те вещи, которые вы намерены проверить в рамках этого плана тестирования. По существу список

что нужно протестировать. Это может быть разработано из целей тестирования программного обеспечения

кадастров, а также других источников документации и информации, таких как:

• Требования к характеристикам

• Технические характеристики

• Руководство пользователя

• Руководства по эксплуатации или руководства

• Инструкции по установке или процедуры

Это можно контролировать и определять с помощью локального управления конфигурацией (CM)

если у вас есть. Эта информация включает номера версий, конфигурацию

при необходимости, (особенно, если поддерживаются несколько версий продукта). Это

могут также включать ключевые вопросы расписания доставки для критических элементов.

Определите любые критические шаги, необходимые до начала тестирования, например, как получить

необходимый предмет.

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

прикладной или функциональной области, для более низких уровней это может быть программа, блок, модуль или сборка.

Следует также включить ссылки на существующие отчеты об инцидентах или запросы на повышение.

В этом разделе также могут быть указаны элементы, которые будут исключены из тестирования

**Тестирование функций – Features to be tested**

Это список того, что должно быть проверено с точки зрения USERS, о ​​том, что система

делает. Это не техническое описание программного обеспечения, а представление USERS о функциях. Это

рекомендуется определить спецификацию дизайна теста, связанную с каждой функцией или набором

функции.

Установите уровень риска для каждой функции. Используйте простую рейтинговую шкалу, такую ​​как (H, M, L); Высокая,

Средний и низкий. Эти типы уровней понятны пользователю. Вы должны быть

чтобы обсудить, почему был выбран конкретный уровень.

Это еще одно место, где инвентаризация целей тестирования может использоваться, чтобы помочь идентифицировать

совокупности целей, подлежащих тестированию вместе (это использует иерархию теста

цели). В зависимости от уровня плана тестирования конкретные атрибуты (цели) объекта или

набор признаков может быть идентифицирован.

**Особенности не подлежат проверке – Features not to be tested**

Это список того, что НЕ должно быть проверено с точки зрения пользователей как

системы и управления конфигурацией / версией управления версиями. Это не технический

описание программного обеспечения, но пользовательский вид функций.

• Определите ПОЧЕМУ функцию не тестировать, может быть любое количество причин.

• Не включаться в этот выпуск Программного обеспечения.

• Низкий риск, использовался до и считается стабильным.

• Будет выпущен, но не проверен или задокументирован как функциональная часть выпуска

эта версия программного обеспечения.

**Подход - Approach**

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

план (хозяин, прием и т. д.) и должен быть согласен со всеми более высокими и более низкими

уровней планов. Необходимо определить общие правила и процессы.

• Используются ли какие-либо специальные инструменты и каковы они?

• Будет ли инструмент требовать специальной подготовки?

• Какие показатели будут собираться?

• На каком уровне должна быть собрана каждая метрика?

• Как управлять конфигурацией?

• Сколько различных конфигураций будет проверено

• Аппаратное обеспечение

• Программного обеспечения

• Комбинации HW, SW и других пакетов поставщиков

• Каковы правила тестирования регрессии? Сколько будет сделано и сколько при каждом испытании

уровень.

• Будет ли регрессионное тестирование основано на обнаружении серьезности обнаруженных дефектов?

• Как элементы в требованиях и дизайне, которые не имеют смысла или являются

не подлежат обработке?

• Если это мастер-план тестирования, общий подход к тестированию проектов и охват

требования также должны быть определены.

• Укажите, есть ли особые требования к тестированию.

• Будет проверен только полный компонент.

• Определенный сегмент группировки компонентов / компонентов должен быть протестирован вместе.

Другая информация, которая может быть полезна при настройке подхода:

• MTBF, среднее время между отказами - если это допустимое измерение для теста

и если данные доступны.

• SRE, Engineering Reliability Engineering - если эта методология используется, и если

информация доступна.

• Как будут проводиться совещания и другие организационные процессы.

• Существуют ли какие-либо существенные ограничения для тестирования.

• Доступность ресурсов

• Сроки

• Существуют ли какие-либо рекомендуемые методы тестирования, которые следует использовать, если да, то почему?

**Критерий прохода / отказа элемента - Item Pass/Fail Criteria**

Каковы критерии завершения этого плана? Это критический аспект любого плана тестирования

и должны соответствовать уровню плана. Цель состоит в том, чтобы определить, является ли тест

элемент прошел тестовый процесс.

• На уровне тестирования модуля это могут быть такие элементы, как:

• Все тестовые примеры завершены.

• Указанный процент случаев, заполненных процентом, содержащим некоторые

количество мелких дефектов.

• Инструмент покрытия кода показывает весь охваченный код.

• На уровне мастер-плана это могут быть такие предметы, как:

• Все планы нижнего уровня завершены.

• Указанное количество планов завершено без ошибок и процент с

незначительные дефекты.

• Это может быть индивидуальный критерий уровня теста или план уровня единицы, или он может быть

общие функциональные требования для планов более высокого уровня.

• Каково количество и серьезность 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 может быть приемлемым для

заявление.

• Сбой - это результат дефекта, который видит пользователь, сбоя системы и т. Д.

**Критерии приостановки и требования к возобновлению - Suspension Criteria and Resumption Requirements**

Знайте, когда нужно приостановить серию тестов или, возможно, завершить набор тестов. После тестирования

приостановлено, как оно возобновляется и каковы потенциальные воздействия (например, регрессионные тесты).

Если число или тип дефектов достигает точки, в которой последующее тестирование не имеет значения,

нет смысла продолжать тест; вы просто тратите ресурсы. Укажите, что представляет собой остановку для теста или серии тестов, и что такое

допустимый уровень дефектов, которые позволят тестированию пройти через дефекты.

• Тестирование после поистине фатальной ошибки создаст условия, которые могут быть идентифицированы как

дефекты, но на самом деле являются ошибками призраков, вызванными более ранними дефектами, которые были проигнорированы.

**Результаты тестирования - Test Deliverables**

Что должно быть сделано в рамках этого плана?

• План тестирования

• Спецификации дизайна испытаний.

• Спецификации тестовых случаев

• Спецификации процедуры испытаний

• Отчеты об отправке тестовых элементов

• Тестовые журналы

• Отчеты об инцидентах

• Сводные отчеты об испытаниях

• Отчеты об инцидентах

Данные испытаний также можно рассматривать как поставляемые, а также возможные инструменты тестирования для

процесс тестирования

Одна вещь, которая не является пробной версией, - это программное обеспечение; который указан в тестовых

поставляемых разработкой.

Эти элементы должны быть определены в общем плане проекта в качестве результатов (этапов)

и должны иметь соответствующие ресурсы, назначенные им в системе отслеживания проектов.

Это обеспечит прозрачность процесса тестирования в рамках всего процесса отслеживания проекта

и что тестовые задачи для создания этих результатов запускаются в соответствующее время. Любые

зависимости между этими результатами и их соответствующим программным обеспечением должны быть

определены. Если документ предшественника является неполным или нестабильным, тестовые продукты будут страдать

также.

**Задачи тестирования - Test Tasks**

Должны быть определены задачи для каждого теста. Включить все межзадачные

зависимости, уровни навыков и т. д. Эти задачи также должны иметь соответствующие задачи и

контрольные точки в общем процессе отслеживания проекта (инструмент).

Если это многофазный процесс или если приложение должно быть выпущено с шагом

могут быть частью приложения, которое этот план не адресует. Эти области должны быть

чтобы избежать какой-либо путаницы, следует сообщить о дефектах в отношении этих будущих функций.

Это также позволит пользователям и тестировщикам избегать неполных функций и предотвращать

преследование ресурсов Непреломления.

Если проект разрабатывается как многопартийный процесс, этот план может охватывать только

часть общих функций / функций. Это необходимо определить, чтобы эти другие области

разработали для них планы и не тратьте средства на отслеживание дефектов ресурсов, которые не

относятся к этому плану.

Когда сторонняя сторона разрабатывает программное обеспечение, этот раздел может содержать описания

те задачи теста, принадлежащие как внутренним группам, так и внешним группам.

**Экологические потребности - Environmental Needs**

Существуют ли какие-либо особые требования для этого плана тестирования, такие как:

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

• Как будут представлены данные тестирования. Существуют ли специальные требования к коллекции или конкретные

диапазоны данных, которые должны быть предоставлены?

• Сколько будет проведено тестирование для каждого компонента многочастной функции?

• Специальные требования к мощности.

• Конкретные версии другого поддерживающего программного обеспечения.

• Ограниченное использование системы во время тестирования.

• Инструменты (как приобретенные, так и созданные).

• Связь

• Веб

• Клиент / Сервер

• Сеть

• Топология

• Внешние

• Внутренний

• Мосты / Маршрутизаторы

• Безопасность

**Обязанности - Responsibilities**

Кто отвечает? Должен быть ответственный человек за каждый аспект тестирования и

процесс тестирования. Каждая идентифицированная тестовая задача также должна быть назначена ответственным лицом.

Это включает в себя все области плана, вот несколько примеров.

• Установка рисков.

• Выбор функций для тестирования и тестирования.

Настройка общей стратегии для этого уровня плана.

• Обеспечение наличия всех необходимых элементов для тестирования.

• Обеспечение разрешения конфликтов планирования, особенно если тестирование выполняется на

производственная система.

• Кто обеспечивает необходимую подготовку?

• Кто принимает критические решения go / no going для предметов, не охваченных планами тестирования?

• Кто поставляет каждый элемент в разделе тестовых элементов?

**Кадровое обеспечение и потребности в обучении - Staffing and Training Needs**

Определите все критические требования и требования к обучению.

• Обучение по продукту.

Обучение использованию любых инструментов тестирования.

**График - Schedule**

Должно быть основано на реалистичных и проверенных оценках. Если оценки для разработки

приложения являются неточными, весь план проекта будет проскальзывать, а тестирование является частью

общий план проекта.

Поскольку мы все знаем, что первая область плана проекта должна быть сокращена, когда дело доходит до времени хруста

окончание проекта - это тестирование. Обычно это сводится к решению: «Поставим что-нибудь

даже если на самом деле это не так хорошо работает ». И, как мы все знаем, это, как правило, худшее

возможно решение.

• Как проскальзывание в расписании должно быть обработано также должно быть рассмотрено здесь.

• Если пользователи заранее знают, что проскальзывание в процессе разработки приведет к проскальзыванию

в тесте и общей доставке системы они могут быть немного более толерантными

если они знают, что в их интересах получить лучшее тестируемое приложение.

• Изложив здесь эффекты, у вас есть изменения, чтобы обсудить их перед их

фактическое происшествие. Вы даже можете заставить пользователей согласиться на несколько недостатков заранее, если

расписание.

• На этом этапе все соответствующие этапы должны быть идентифицированы с их отношением к

выявлен процесс разработки. Это также поможет в определении и отслеживании

потенциальное проскальзывание в расписании, вызванное процессом тестирования.

• Лучше всего привязывать все даты тестирования непосредственно к связанным с ними датам развития.

Это препятствует тому, чтобы тестовая команда воспринималась как причина задержки. Для

Например: если тестирование системы начнется после доставки окончательной сборки, тогда система

тестирование начинается через день после родов. Если доставка заканчивается, тестирование системы начинается с

день доставки, а не на определенную дату.

Для оценки усилий, необходимых для тестирования, необходимо учитывать множество элементов. Это

критически важно, чтобы как можно больше информации входила в оценку как можно скорее в

чтобы обеспечить точное планирование тестирования. Для обобщенного списка соображений см.

документ оценки в Приложении C.

**Риски и непредвиденные расходы - Risks and Contingencies**

Каковы общие риски для проекта с акцентом на процесс тестирования?

• Отсутствие кадровых ресурсов, когда тестирование начнется.

• Отсутствие наличия необходимого оборудования, программного обеспечения, данных или инструментов.

• Поздняя доставка программного обеспечения, оборудования или инструментов.

• Задержки в обучении по применению и / или инструментам.

Изменения первоначальных требований или конструкций

• Укажите, что будет сделано для различных событий, например:

• Определение требований будет завершено до 1 января 19XX, и если

изменения требований после этой даты будут предприняты следующие действия.

• График тестирования и график разработки выведут соответствующее количество

дней. Это редко происходит, поскольку большинство проектов имеют фиксированные сроки поставки.

• Количество выполненных тестов будет уменьшено.

• Количество допустимых дефектов будет увеличено.

• Эти два элемента могут снизить общее качество поставляемого продукта.

• Ресурсы будут добавлены в тестовую группу.

• Команда тестирования будет работать сверхурочно.

• Это может повлиять на моральный дух команды.

• Сфера действия плана может быть изменена.

• Возможно, будет оптимизация ресурсов. Этого следует избегать, если это возможно

по очевидным причинам.

• Вы можете просто ВЫЙТИ. Довольно экстремальный вариант, если не сказать больше.

Обычно руководство неохотно принимает сценарии, такие как выше, хотя

они видели, как это происходит в прошлом. Важно помнить, что если вы это сделаете

ничего не получается, обычный результат заключается в том, что тестирование полностью или полностью исключено, ни одно из

который должен быть приемлемым вариантом

**Сертификаты - Approvals**

Кто может одобрить процесс как полный и позволить проекту перейти к следующему

(в зависимости от уровня плана).

• На уровне основного тестового плана это могут быть все вовлеченные стороны.

• При определении процесса утверждения имейте в виду, кто аудитория.

• Аудитория для плана уровня единичного тестирования отличается от аудитории интеграции,

системы или уровня мастер-уровня.

• Уровни и тип знаний на разных уровнях будут разными.

• Программисты очень технические, но могут не иметь четкого понимания

общий бизнес-процесс, управляющий проектом.

• Пользователи могут иметь различные уровни деловой хватки и очень мало технических навыков.