

Описание процессов жизненного цикла программного обеспечения

«SDK onlineconvertfree – облачный конвертер файлов»

Содержание

Аннотация	3
Введение	4
1. Планирование процессов жизненного цикла разработки	5
2. Формирование и оценка задач	7
3. Проектирование и разработка	8
4. Тестирование и отладка	9
5. Эксплуатация и сопровождение	11
5.1. Техническая поддержка	11
5.2. Персонал, обеспечивающий работу на местах	12
5.3. Персонал, обеспечивающий техническую поддержку и развитие системы	12
6. Совершенствование программного обеспечения	13
Приложение А. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ТЕРМИНОВ	14
Приложение В. Инструкция подключения	15

Аннотация

Настоящий документ является описанием процессов жизненного цикла программного обеспечения «SDK onlineconvertfree – облачный конвертер файлов» и содержит сведения о жизненном цикле программного обеспечения, в том числе устранение неисправностей, выявленных в ходе эксплуатации программного обеспечения, а также информацию о персонале, необходимом для обеспечения такой поддержки.

Введение

Основным направлениями деятельности ООО «Вебпрактик» являются: разработка и производство автоматизированных информационных систем, а также сопровождение их функционирования.

Одним из продуктов ООО «Вебпрактик» является облачный сервис <https://onlineconvertfree.com/>, включающий в себя программное обеспечение «SDK onlineconvertfree – облачный конвертер файлов».

Основными процессами жизненного цикла программной продукции являются:

- формирование и оценка задач;
- проектирование и разработка;
- тестирование и отладка;
- эксплуатация и сопровождение.

1. Планирование процессов жизненного цикла разработки

Жизненный цикл (ЖЦ) включает период создания и использования «SDK onlineconvertfree – облачный конвертер файлов», начиная с момента возникновения потребности в продукте, заканчивая разработкой, тестированием и отладкой, поставкой программной продукции, ее эксплуатацией на объектах Заказчика и технической поддержкой.

Жизненный цикл определен с учетом положений следующих стандартов:

- ГОСТ Р ИСО/МЭК 12207:2010 «Информационная технология. Системная программная инженерия. Процессы жизненного цикла программных средств»;
- ГОСТ Р 56939-2016 «Защита информации. Разработка безопасного программного обеспечения. Общие требования»;
- ГОСТ РВ 0015002:2012 «Системы менеджмента качества. Общие требования».

В ООО «Вебпрактик» принята итерационная модель жизненного цикла.

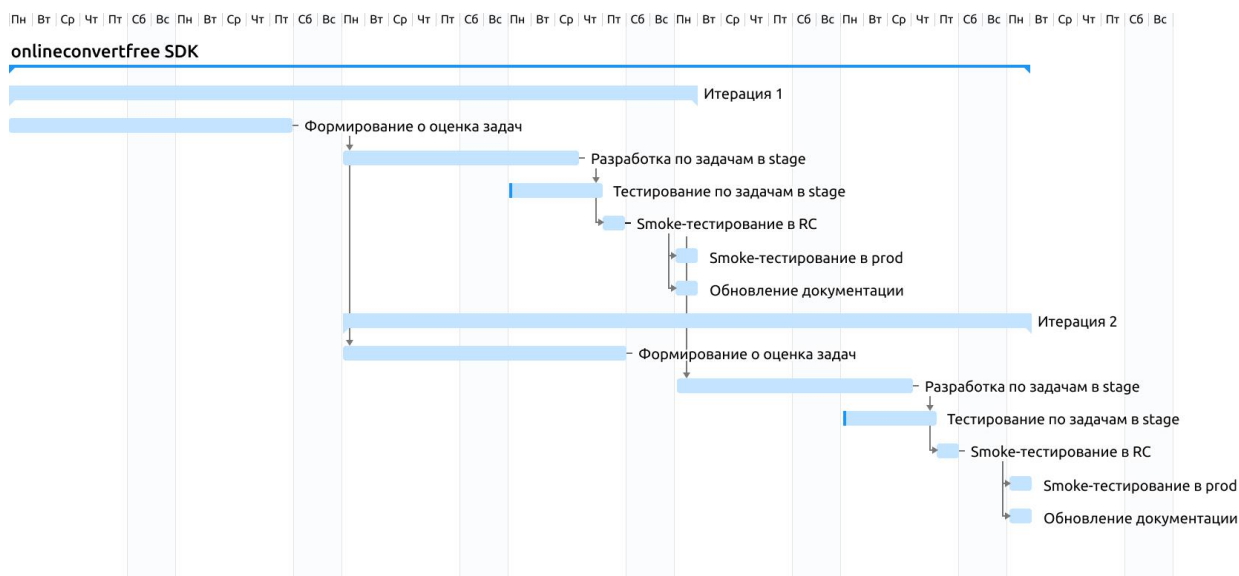


Рисунок №1. Модель жизненного цикла ПО

Разработка продукта ведется по итеративной модели с выпуском новой версии в среднем каждые две недели. Каждая итерация включает в себя этапы формирования и оценки задач, разработки и тестирования по задачам, smoke тестирования и обновления документации.

Модель жизненного цикла обеспечивает необходимый контроль над разработкой и сопровождением ПО «SDK onlineconvertfree – облачный конвертер файлов».

В ООО «Вебпрактик» процессы управления конфигурацией сервиса осуществляются с использованием репозитория дистрибутивов, стенда сборки и системы контроля версий.

2. Формирование и оценка задач

Процессом формирования и оценки задач определен порядок:

- ответственный за продукт аналитик определяет задачи, которые должны быть выполнены в итерацию;
- ответственный за продукт аналитик определяет требования к задачам;
- технический менеджер, ответственный за продукт аналитик совместно с командой разработки проводят оценку задач;
- технический менеджер продукта устанавливает ответственность за разработку, распределяя задачи между членами команды разработки;
- технический менеджер продукта осуществляет мониторинг этого процесса (установлены критерии и методы оценки результативности процесса).

Управление задачами ведётся с использованием выпусков (далее, по тексту – релизов), т.е. все задачи, запланированные в релиз, помечаются соответствующим образом. Задачи, не включенные в какой-либо релиз, остаются в списке задач по продукту (далее, по тексту – бэклог) и не участвуют в разработке, а используются только для планирования будущих работ.

Результатами этапа являются:

1. Сформирован бэклог на итерацию и релиз;
2. Подготовлена спецификация требований и/или техническое описание по каждой задаче;
3. Назначен исполнитель по каждой задаче.

3. Проектирование и разработка

Процессом формирования и оценки задач определен порядок:

- разработка программной архитектуры, а также разработка решений по построению всех составных компонент;
- разработка исходных текстов, написание файлов спецификации для сборки пакетов прикладного программного обеспечения;
- сборка пакетов прикладного программного обеспечения и добавление их в репозиторий программного обеспечения;
- сборка дистрибутивов из репозитория программного обеспечения.

Разработка по задачам идёт до тех пор, пока все задачи в рамках итерации не будут закрыты.

Так, в каждом репозитории есть три основных ветки и соответствующих им окружения:

- Стейдж (с англ. Develop, в терминологии гитфлоу) – базовая ветка для разработки, от неё идут все ветки задач (feature ветки);
- RC (с англ. Release Candidate, в терминологии гитфлоу) – ветка готовящегося релиза;
- Мастер (с англ. Master, в терминологии гитфлоу) – ветка последнего стабильного релиза.

4. Тестирование и отладка

Проведение тестирования является обязательным перед передачей новой версии потребителю. Тестирование проводится лицами, ответственными за проведение тестирования программной продукции (отдел тестирования). Для тестирования и отладки программной продукции собирается стенд, выдается задание на тестирование. По результатам тестирования осуществляется устранение ошибок и осуществляется (при необходимости) доработка программного обеспечения.

Процессом тестирования и отладки определен порядок:

- сборка дистрибутивов программного обеспечения - ответственный за продукт инженер по внедрению проводит слияние **stage** в **RC** и сборку в **RC** окружении;
- проведение тестирования программного обеспечения – ответственный за продукт инженер-тестировщик проводит тестирование в **RC**, а также дымовое тестирование (с англ. smoke testing) в **RC** всего продукта;
- устранение выявленных недостатков программного обеспечения – в случае обнаружения недостатков инженер-тестировщик формирует задачу с описанием дефекта (недостатка ПО), далее проводится повторное дымовое тестирование (ретест);
- при завершение дымового тестирования (в том числе повторного) проводится регрессионное тестирование всей функциональности продукта;
- при положительном результате регрессионного тестирования проводится добавление в репозиторий эталонных версий дистрибутивов и исходных текстов программного обеспечения;
- ответственный за продукт инженер по внедрению проводит слияние **RC** в **master** и сборку в **prod** окружении;
- корректировка программной документации.

По итогам тестирования и отладки сотрудник, ответственный за выпуск новой версии формирует заключение о качестве версии с оценкой уровня исправления ошибок и запускает процесс согласования разрешения на выпуск версии со следующими лицами:

- сотрудник проектной группы, ответственный за разработку ПО «SDK onlineconvertfree – облачный конвертер файлов»;
- руководитель проекта;
- руководитель подразделения, утверждающий разрешение на выпуск версии ПО «SDK onlineconvertfree – облачный конвертер файлов».

5. Эксплуатация и сопровождение

5.1. Техническая поддержка

Техническая поддержка пользователей осуществляется в формате консультирования пользователей и администраторов сервиса по вопросам установки, переустановки, администрирования и эксплуатации программного обеспечения по электронным каналам связи (электронной почте support@onlineconvertfree.com или на web-портале <https://onlineconvertfree.com/>). В рамках технической поддержки сервиса оказываются следующие услуги:

- помощь в установке программного обеспечения;
- помощь в настройке и администрировании программного обеспечения;
- помощь в установке обновлений программного обеспечения;
- помощь в поиске и устранении проблем в случае некорректной установки обновления программного обеспечения;
- описание функционала программного обеспечения, помощь в эксплуатации;
- предоставление актуальной документации по установке/настройке/работе программного обеспечения.

В заявке на техническую поддержку пользователь должен указать следующую информацию:

- описание проблемы;
- предпринятые попытки решения проблемы;
- любая релевантная дополнительная информация.

После доставки ответа пользователю запрос считается завершенным, и находится в статусе «завершен, требует подтверждения пользователя». В случае аргументированного несогласия пользователя с завершением запроса, выполнение запроса продолжается.

Завершенный запрос переходит в состояние закрытого после получения подтверждения от пользователя о решении запроса. В случае отсутствия ответа пользователя о завершении запроса в течение 14 рабочих дней, в случае если иное не оговорено в соглашении о

расширенной технической поддержки, запрос считается закрытым. Закрытие запроса может инициировать пользователь, если надобность в ответе на запрос по каким-либо причинам более не требуется.

5.2. Персонал, обеспечивающий работу на местах

Пользователи ПО «SDK onlineconvertfree – облачный конвертер файлов» должны обладать навыками работы с персональным компьютером на уровне пользователя. Для работы с Системой пользователю необходимо изучить руководство пользователя «SDK onlineconvertfree – облачный конвертер файлов».

Администратор Системы должен владеть навыками работы с персональным компьютером на уровне уверенного пользователя. Обязательно знание основ работы вычислительной техники и программного обеспечения в локальных сетях, а также настроек системной политики прав пользователей в операционных системах семейства Windows.

5.3. Персонал, обеспечивающий техническую поддержку и развитие системы

Специалисты, обеспечивающие техническую поддержку и развитие Системы, должны обладать следующими знаниями и навыками:

Владение персональным компьютером на уровне уверенного пользователя;

1. Знание функциональных возможностей Системы и особенностей работы с ними;
2. Знание языков программирования;
3. Знание реляционных БД;
4. Знание протоколов обмена данных;
5. Знание средств восстановления баз данных и мониторинга производительности серверов.

Коллектив разработчиков (программисты, консультанты, технические специалисты) обладают необходимым набором знаний для работы со всеми компонентами, входящими в состав ПО, при решении прикладных задач, соответствующих функционалу программ.

6. Совершенствование программного обеспечения

Процесс совершенствование программного обеспечения осуществляется с использованием двух методов:

- формирование списка гипотез по увеличению качества программного обеспечения;
- формирования списка задач по результатам анализа обращений в службу поддержки.

Ответственный за проект технический менеджер формирует список гипотез по увеличению качества программного обеспечения. Ответственный за проект аналитик проводит анализ обращений пользователей в службу поддержки. В результате этих двух процессов формируется список задач, направленных на совершенствование программного обеспечения. Задачи приоритизируются по трем параметрам: критичность, вера в эффективность и сложности реализации. Задачи с высокой критичностью включаются в очередную итерацию разработки в первую очередь (см. п. 2). Задачи с высокой верой в эффективность и простой реализацией формируют очередь из задач на реализацию в очередной итерации разработки. Остальные задачи реализацию в рамках свободных ресурсов команды разработки со средним приоритетом.

Приложение А. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ТЕРМИНОВ

Бэклог продукта	упорядоченный набор элементов, очередь задач, перечень всех функций, которые заинтересованные люди хотят получить от продукта
Релиз	выпуск программы/кода/библиотеки продукта, готового для использования
Гитфлоу	методология работы с Git, то есть в ней определяется, какие виды веток необходимы проекту и как выполнять слияние между ними
Git	распределенная система контроля версий и управления исходным кодом с упором на скорость
Дымовое тестирование	в тестировании программного обеспечения означает минимальный набор тестов на явные ошибки
Регрессионное тестирование	это исследование, испытание программного обеспечения (иными словами, тестирование), направленное на обнаружение ошибок в уже проверенных участках программ (или исходных кодах)

Приложение В. Инструкция подключения

1. Подключиться по SSH к тестовому серверу командой `ssh sdk@62.109.9.105`.

При запросе пароля ввести пароль `*$j@i0=+urfQ-#)n`

2. Перейти в каталог с экземпляром скрипта, использующим SDK, командой `cd ocf-converter-sdk-php-example`
3. Выполнить тестовый запуск скрипта.

Для выполнения скрипта используется команда `php index.php` с параметрами:

`--apiKey` - параметр для передачи ключа API. Ключ можно получить на странице <https://onlineconvertfree.com/file-conversion-api/> после регистрации, либо использовать тестовый ключ `4d1344e229fe8c646090c3b88bd1a11b`

`--filePath` - путь к файлу для конвертации. Можно использовать тестовый файл `/home/sdk/ocf-converter-sdk-php-example/example.png`

`--to` - расширение формата, в который нужно конвертировать файл, из поддерживаемых сервисом <https://onlineconvertfree.com/>. Список поддерживаемых форматов указан на странице <https://onlineconvertfree.com/file-format/>

Для файлов `png` можно использовать при тестировании расширение `pdf`.

Пример команды для тестового запуска:

```
php index.php --apiKey=4d1344e229fe8c646090c3b88bd1a11b --filePath=./example.png --to=pdf
```

4. При успешном выполнении скрипта в каталоге появится файл с именем `result` и расширением, запрошенным при запуске команды, например, `result.pdf`, можно увидеть этот файл, выполнив команду `ls result.pdf`.
5. Закрывать подключение командой `exit`.
6. При необходимости можно скачать файл `result.pdf`, выполнив на своей машине команду `scp sdk@62.109.9.105:/home/sdk/ocf-converter-sdk-php-example/result.pdf result.pdf`, введя при запросе тот же пароль `*$j@i0=+urfQ-#)n`, после чего можно открыть файл, чтобы увидеть его содержимое.