

Описание процессов жизненного цикла программного обеспечения

«Программное обеспечение «Система поддержки принятия врачебных решений Предсказание осложнений пациента с пневмонией, включая вирусную пневмонию, по данным ЭМК»»

Версия 1.0.0

Аннотация

Настоящий документ является описанием процессов жизненного цикла «Программное обеспечение «Система поддержки принятия врачебных решений Предсказание осложнений пациента с пневмонией, включая вирусную пневмонию, по

данным ЭМК» 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.

Содержание

Аннотация	1
Содержание	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	10
5.1. Техническая поддержка	10
5.2. Персонал, обеспечивающий работу на местах	10
5.3. Персонал, обеспечивающий техническую поддержку и развитие системы	11
Приложение А	12

Введение

Основными направлениями деятельности ООО «Сбермедии» являются: разработка и производство автоматизированных медицинских систем на базе искусственного интеллекта (далее, по тексту - ИИ), сопровождение их функционирования, а также тематические исследования в области машинного обучения и анализа данных в медицине.

Одним из продуктов ООО «Сбермедии» является «Программное обеспечение «Система поддержки принятия врачебных решений Предсказание осложнений пациента с пневмонией, включая вирусную пневмонию, по данным ЭМК»».

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 «Программное обеспечение «Система поддержки принятия врачебных решений Предсказание осложнений пациента с пневмонией, включая вирусную пневмонию, по данным ЭМК»», начиная с момента возникновения потребности в продукте, заканчивая разработкой, тестированием и отладкой, поставкой программной продукции, ее эксплуатацией на объектах Заказчика (ЛПУ) и технической поддержкой.

Жизненный цикл определен с учетом положений следующих стандартов:

- ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»;

- ГОСТ Р 56939-2016 «Защита информации. Разработка безопасного программного обеспечения. Общие требования»;

- ГОСТ РВ 0015-002-2012 «Системы менеджмента качества. Общие требования». В ООО «Сбермеди» принята итерационная модель жизненного цикла (рисунок 1)

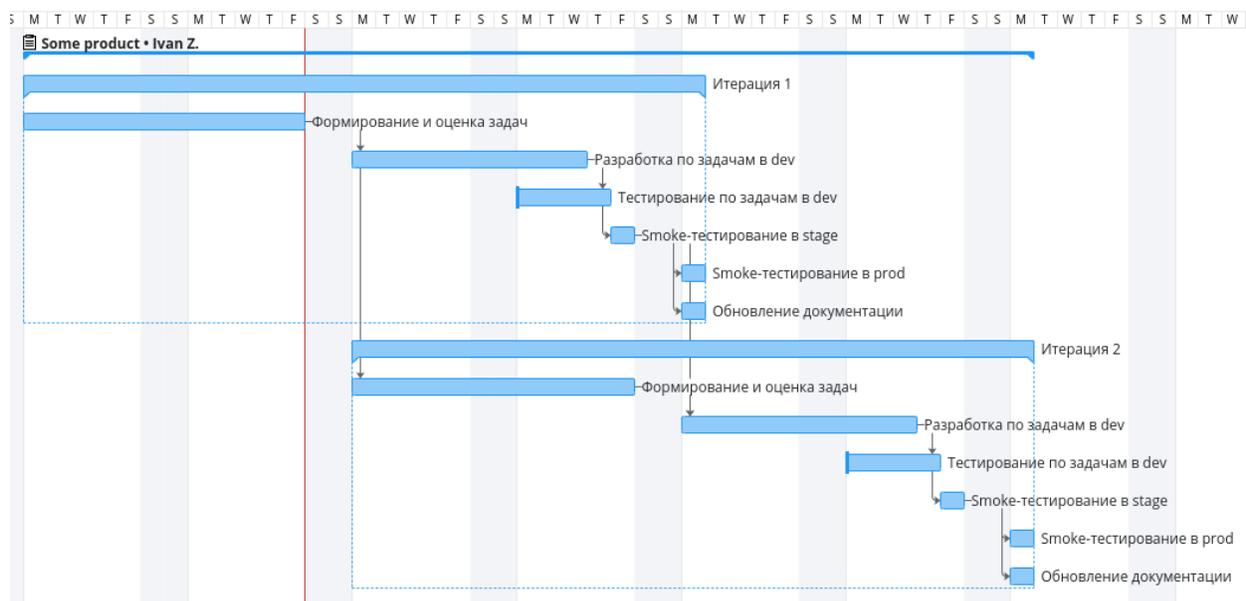


Рисунок №1. Модель жизненного цикла ПО

Разработка продукта ведётся 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 «Программное обеспечение «Система поддержки принятия врачебных

решений Предсказание осложнений пациента с пневмонией, включая вирусную пневмонию, по данным ЭМК»».

В ООО «Сбермедии» 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. Именование версий происходит по методологии semver - <https://semver.org/lang/ru>. 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.

При работе с кодом команда придерживается методологии, известной как гитфлоу (<https://nvie.com/posts/a-successful-git-branching-model>), но с несколькими упрощениями.

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

- Дев (с англ. Develop, в терминологии гитфлоу) - базовая ветка для разработки, от неё идут все ветки задач (feature ветки);
- Стейдж (с англ. Stage, 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 **develop** в **stage** и сборку в **stage** окружении;
- проведение тестирования программного обеспечения - ответственный за продукт инженер-тестировщик проводит тестирование в **stage**, а также дымовое тестирование (с англ. smoke testing) в **stage** 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 (ре-тест);
- 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 **stage** в **master** с проставлением тега релиза (х.у.з) 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 «Программное обеспечение «Система поддержки принятия врачебных решений Предсказание осложнений пациента с пневмонией, включая вирусную пневмонию, по данным ЭМК»»;
- руководитель проекта;
- руководитель подразделения, утверждающий разрешение на выпуск версии «Программное обеспечение «Система поддержки принятия врачебных решений Предсказание осложнений пациента с пневмонией, включая вирусную пневмонию, по данным ЭМК»».

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 helpdesk@sbermed.ai или на web-портале helpdesk.sbermed.ai). 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 поддержке, 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 «Программное обеспечение «Система поддержки принятия врачебных решений Предсказание осложнений пациента с пневмонией, включая вирусную пневмонию, по данным ЭМК»» должны обладать навыками работы с персональным компьютером на уровне пользователя. Для работы с Системой пользователю необходимо

изучить руководство пользователя «Программное обеспечение «Система поддержки принятия врачебных решений Предсказание осложнений пациента с пневмонией, включая вирусную пневмонию, по данным ЭМК»».

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:

1. Владение персональным компьютером на уровне уверенного пользователя;
2. Знание функциональных возможностей Системы и особенностей работы с ними;
3. Знание языков программирования;
4. Знание реляционных БД;
5. Значение протоколов обмена данных;
6. 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.

Приложение А

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ТЕРМИНОВ

ЛПУ	- лечебно-профилактическое учреждение
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)