

ДОРИС Реакт

Описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения, способов устранения неисправностей, информации о персонале, информации о совершенствовании ПО.

1. Процесс разработки и доработки программного обеспечения ДОРИС Реакт (react) включает в себя следующие основные этапы:

- формирование функциональных требований;
- проектирование и формирование спецификации требований к компонентам системы;
- разработка (доработка) ПО в соответствии со сформированными спецификациями требований;
- тестирование функций разработанного (доработанного) ПО на стенде;
- устранение замечаний по итогам тестирования (в случае, если они были выявлены);
- корректировка документации (если доработка влечет изменение документированных функций);
- установка актуальной версии программного обеспечения.

В качестве модели разработки (доработки) программного обеспечения применяется гибкая модель. Команда разработки совместно планирует очередной этап (обсуждает планируемые функции, которые могут быть реализованы за выделенное время, определяет возможные варианты решения и распределяет задачи по участникам) с целью добавить функционал в очередном релизе. Данная методология позволяет эффективно использовать трудовой ресурс и обеспечить быструю обратную связь от задействованных участников. В ходе работы команда использует систему управления проектами для планирования и контроля выполнения задач и, систему контроля версий при совместной работе для хранения версий программного кода

1.1 В процессе разработки (и модернизации) принимают участие специалисты:

- 1 аналитик;
- 1 веб-программист;
- 1 инженер-программист;
- 1 администратор баз данных;
- 1 тестировщик.

1.2 Разработка ПО проводится специалистами компании ООО «НПО «ИТС СОФТ» по адресу: 119334, город Москва, 5-Й Донской проезд, д. 15 стр. 2, этаж 2 помещ./ком. 2/IV/31а

1.3 Контактные данные:

- Телефон: +7 495 796 13 86
- электронная почта: info@npo-its.ru

2. Процессы технической поддержки и гарантийного обслуживания

программного обеспечения «ДОРИС Реакт». Цель процесса поддержки и гарантийного обслуживания заключается в обеспечении гарантии качества оказанных услуг по контракту (или договору) и того, что по всем поступающим обращениям в части функционирования программного обеспечения предоставляется решение или консультация.

В процессе технической поддержки, гарантийного обслуживания и решения проблем:

- а) обращения регистрируются в системе управления запросами (СУЗ);
- б) запросы анализируются и оцениваются для определения приемлемого способа решения проблемы;
- в) определяется способ решения проблемы (доработка программного обеспечения, настройка программного обеспечения, консультация по работе в программе или дополнительно определяется контрактом);
- г) реализация принятого способа решения проблемы.

2.1 Режим работы службы тех.поддержки и гарантийного обслуживания- с 9:00 до 18:00 (Московское время).

2.2 Прием и обработка заявок - режим и способ (почта, СУЗ) по условиям контракта (или договора);

2.3 В процессе тех.поддержки и гарантийного обслуживания задействовано 2 инженера службы поддержки авторизованной сервисной службы. В случае, если требуется доработка программного обеспечения, также задействуются специалисты команды разработки;

2.4 Фактический почтовый адрес, по которому осуществляется процесс сопровождения - 119334, город Москва, 5-Й Донской проезд, д. 15 стр. 2, этаж 2 помещ./ком. 2/IV/31а

3. Для установки и настройки программного обеспечения, а также осуществления поддержки функционирования программного обеспечения специалисты должны обладать следующими навыками:

- Администрирование ОС Linux;
- Администрирование баз данных PostgreSQL;
- Администрирование ОС Windows Server;
- Настройка серверов приложений Microsoft IIS;
- Настройка сетевого взаимодействия.

4. Совершенствование и модернизация программы происходит путем доработки программного обеспечения с целью информационного взаимодействия со специализированными информационными системами, оптимизации производительности программного обеспечения, добавления нового функционала (сбор, хранение, защита, предоставление и распространение информации) для выполнения сценарного плана. Изменение программного обеспечения сопровождается изменением номера версии.

5. В процессе тестирования и эксплуатации программное обеспечение может в интерфейсе выводить диагностические сообщения о неисправности или фиксироваться некорректное выполнение документированных функций. В таких случаях необходимо осуществить процедуру передачи информации о

характере ошибки (и условиях, при которых они возникли) в поддержку авторизованной сервисной службы компании ООО «НПО ИТС» (п. 2.2). Устранение неисправностей и техническое обслуживание может осуществлять только квалифицированный персонал, а именно сотрудники авторизованной сервисной службы компании ООО «НПО «ИТС СОФТ». Для оформления заявки на устранение неисправности необходимо отправить заявку на электронную почту info@proits.ru или почтой по адресу: 119334, город Москва, 5-й Донской проезд, д. 15 стр. 2, этаж 2 помещ./ком. 2/IV/31а. Для разрабатываемого на предприятии ПО в качестве модели жизненного цикла для большинства проектов выбирается спиральная модель, соответствующая масштабу и сложности проекта.

Процессы, операции и задачи выполняемые в ходе жизненного цикла ПО соответствуют требованиям ГОСТ 34.601-90.

Спиральная модель схожа с инкрементной моделью, но в ней уделяется больше внимания оценке и разрешению рисков. Спиральная модель подразделяет реализацию проекта на четыре этапа: планирование проекта, оценка рисков, проектирование и разработка и проведение оценки. Проект каждый раз заново проходит через эти четыре стадии при создании новой версии или фрагмента ПО (что соответствует одному витку спирали в данной модели). Базовый виток спирали, начинающийся на этапе постановки задач, включает в себя определение требований и оценку рисков. Каждый последующий виток строится на основе базового.

Требования к программному продукту и к особенностям разработки программного обеспечения определяются на этапе постановки задач. На этапе оценки и разрешения рисков применяется специальный процесс для определения рисков и нахождения разных решений по их разрешению. По окончании данного этапа создается прототип ПО.

Третий этап включает в себя непосредственно разработку ПО и его тестирование по окончании данного этапа. Этап проверки позволяет разработчику и заказчику оценить результат проекта на текущий момент, прежде чем начнется новый виток разработки.

Каждый этап жизненного цикла ПО подразделяется на различные процессы и операции.

На первом этапе определяется общая концепция разрабатываемого программного продукта («вид с высоты птичьего полета»), и на ее основе строится базовая структура проекта, оценивается его выполнимость и связанные с ним риски, описываются соответствующие подходы к конфигурационному управлению и технологиям.

Наиболее важная часть плана проекта – декомпозиция системы на совокупность составных частей в соответствии с требованиями высокого уровня системной иерархии. Все требования к компонентам ПО, устанавливаемые на этапе определения требований, следуют из одного или нескольких требований высокого уровня.

Результатами этапа проектирования являются план конфигурационного управления, план реализации качества ПО, план реализации проекта и календарный план, содержащий подробный список запланированных работ грядущего этапа определения требований, а также предварительная оценка трудозатрат на последующих этапах.

Определение требований

В процессе определения требований в качестве исходных данных используются цели, поставленные в разделе плана проекта, описывающем требования высокого уровня.

Каждый программный компонент должен иметь собственную Спецификацию требований.

Требования к ПО определяют функционал программного компонента, производительность, точность, временные характеристики работы, затраты ресурсов используемого оборудования, работоспособность в нестандартных условиях и при перегрузках. В Спецификации требований к программному обеспечению описываются алгоритмы и математические методы.

Оценка рисков проекта

Риск – это любое событие, которое может помешать реализации проекта в соответствии с планом или его успешному завершению. Риски можно идентифицировать из разных источников. Некоторые из них могут быть довольно очевидными и будут выявлены до начала проекта. Другие будут идентифицированы в течение жизненного цикла проекта, и риск может быть идентифицирован любым участником проектом. Некоторые риски будут присущи самому проекту, в то время как другие будут результатом внешних воздействий, которые полностью неподконтрольны команде проекта.

Этап проектирования

На этапе проектирования в качестве исходных данных используются требования, определенные в принятой спецификации. По каждому требованию определяется элемент или набор элементов проектирования по результатам согласования с заказчиком, моделирования или работы над прототипами.

Элементы проектирования подробно описывают требуемый функционал ПО и, как правило, включают в себя схемы функциональной иерархии, схемы расположения элементов визуализации на экране, таблицы правил, схемы деловых процессов, псевдокод, а также схему всех потоков данных с полным словарем данных. Эти элементы проектирования предназначены для описания ПО в объеме, достаточном для того, чтобы квалифицированные специалисты могли разработать ПО с минимальной потребностью в дополнительных данных.

Разработка архитектуры

На данном этапе определяется архитектурный проект системы, в соответствии с которым выполняется идентификация элементов ПО и удовлетворяются заданные требования. При определении верхнего уровня архитектуры системы должны быть идентифицированы составные части технических средств, программных средств и ручных операций. Должно быть учтено, что все системные требования распределяются между этими составными частями. Составные части конфигурации технических средств, программных средств и ручных операций должны последовательно идентифицироваться этими составными частями.

Этап разработки ПО

На этапе разработки ПО в качестве исходных данных используются элементы проектирования, описанные в принятом плане разработки. По каждому элементу определяется артефакт или набор артефактов ПО. Артефакты ПО включают в себя (но не ограничиваются ими) меню, диалоговые окна, формы для управления данными, форматы отчетных данных и специализированные процедуры, и функции.