



ВИЗОР

СИСТЕМА ЗАЩИТЫ ИНФОРМАЦИИ

**регламент разработки и внедрения
программного обеспечения**

Аннотация

Настоящий регламент определяет правила организации и проведения работ по разработке и внедрению Системы защиты информации (СЗИ) «Визор».

Основными задачами настоящего регламента являются:

- определение сферы применения;
- описание процессов, участников и схемы их взаимодействия;
- определение требований к процедурам деятельности;
- описание обязанностей участников деятельности.

Настоящий регламент ориентирован на применение объектно-ориентированных технологий и современных средств управления проектами, разработки и поддержки версионности программного обеспечения.

Содержание

Аннотация.....	1
Содержание.....	2
Термины и определения	3
Участники жизненного цикла	6
Этапы жизненного цикла.....	8
Разработка программного продукта.....	9
Контур сбора требований.....	11
Контур среднесрочного планирования.....	12
Контур аналитических работ	12
Контур разработки версии	13
Контур небольших доработок.....	14
Информационная модель процесса.....	15
Внедрение программного продукта.....	19
Предпроектные работы	19
Разработка	20
Внедрение	23
Эксплуатация	24
Сопровождение	25
Утилизация.....	26

Термины и определения

Адаптивное сопровождение - модификация программного продукта, обеспечивающая его работоспособность в измененных или изменяющихся условиях (среде).

Архитектура - совокупность существенных решений по организации программного продукта, а также набор структурных элементов и интерфейсов, из которых она состоит, вместе с поведением, описываемым в терминах коопераций этих элементов. К архитектуре программной системы относятся также использование, функциональность, производительность, гибкость, повторное применение, ясность, экономические и технологические ограничения и компромиссы, а также эстетические аспекты.

Артефакт - элемент информации, используемый или порождаемый в процессе разработки или эксплуатации программного обеспечения.

Бизнес-актер - объект, работающий за пределами системы, но взаимодействующий с ней.

Бизнес-процесс - заданная последовательность событий, осуществляемая группой логически связанных деятельностей, использующих ресурсы организации для получения результата при выполнении целей организации. Бизнес-процесс описывается с помощью бизнес-функций, которые показывают ожидаемое поведение программного обеспечения.

Бизнес-роль - поименованное специфическое поведение сущности в конкретной ситуации.

Бизнес-функция - набор действий при выполнении бизнес-процесса, которые дают полезный результат конкретному бизнес-актеру.

Версия - относительно полный и самосогласованный набор артефактов, предназначенный для внутреннего или внешнего использования.

Задача - готовый для интеграции в общую систему программный элемент, реализуемый в минимальные сроки.

Инкрементный подход - процесс непрерывного развития программного обеспечения, когда каждая новая версия содержит улучшения по сравнению с предыдущей.

Итерационный метод - метод разработки программной системы, при котором осуществляется процесс управления потоком исполняемых версий.

Итерация - четко очерченный перечень работ, для которых определены конечная цель и критерий оценки. В результате нескольких итераций должна быть выпущена версия программного продукта.

Квалификационное тестирование - тестирование, проводимое разработчиком и засвидетельствованное заказчиком в целях демонстрации соответствия программного продукта утвержденному техническому заданию и готовности его использования в рабочих условиях.

Корректирующее сопровождение - реактивное изменение программного продукта, выполняемое после его поставки для корректировки обнаруженных проблем (несоответствий, ошибок). Подобные изменения исправляют программный продукт для того, чтобы он соответствовал установленным требованиям.

Полное сопровождение - модификация программного продукта после поставки для повышения его рабочих характеристик, добавления функциональности или улучшения сопровождаемости. Полное сопровождение обеспечивает модернизацию (усовершенствование) программного продукта в интересах пользователя, уточнение соответствующих программных документов и его перепрограммирование для улучшения рабочих характеристик и других атрибутов программного продукта.

Предложение о модификации (ПМ) - общий термин, используемый для определения предполагаемых изменений в сопровождаемом программном продукте. Конкретное предложение о модификации может быть далее классифицировано как коррекция или модернизация и определено как корректирующий, предупреждающий, адаптивный или полный тип сопровождения.

Предупреждающее сопровождение - модификация программного продукта после поставки в целях обнаружения и корректировки имеющихся в нем скрытых ошибок для предотвращения явного проявления этих ошибок при эксплуатации данного продукта.

Программное обеспечение - совокупность программ обработки информации и программных документов, необходимых для эксплуатации этих программ.

Программный продукт - программная система, имеющая коммерческую ценность, готовая к использованию или уже эксплуатируемая.

Программная система - множество элементов программного обеспечения, организованных для достижения конкретной цели, иногда разложенных на несколько подсистем и описываемых набором моделей, возможно, с различных точек зрения.

Проект - уникальный процесс, состоящий из набора координируемых и управляемых видов деятельности с датами начала и окончания, предпринимаемых для внедрения программного продукта в организации в соответствии с оговоренными в Техническом задании требованиями, включая ограничения по срокам, затратам и ресурсам.

Разработчик - организация, которая осуществляет действия по разработке (включая анализ требований, проектирование, реализацию программного продукта вплоть до квалификационного тестирования) в течение процесса жизненного цикла программного продукта.

Роль - определенное поведение и обязанности личности или индивидуумов, работающих в команде (рабочей группе).

Сообщение о проблеме (СП) - термин, используемый для определения и описания проблем, обнаруженных в программном продукте в процессе сопровождения.

Техническое задание - основной документ Проекта, содержащий описание задачи, цель и способы ее внедрения, а также требования к Программному продукту.

Требование - желаемые функциональность, свойство или поведение объекта (системы).

Этап - перечень процессов, которые необходимо выполнить за промежуток времени между двумя опорными точками в жизненном цикле программного продукта, в течение которого должны быть достигнуты заранее поставленные четко определенные цели, артефакты доведены до готовности и принято решение об окончании данного этапа или переходе к следующему.

Участники жизненного цикла

При выполнении процессов жизненного цикла программного продукта определяются следующие основные участники (роли):

Участник	Зона ответственности
Заказчик	<ul style="list-style-type: none">отдельное лицо или организация, заинтересованные в использовании или изменении существующего либо создании нового функционала программного продукта
Пользователь	<ul style="list-style-type: none">отдельное лицо или организация, которые эксплуатируют программный продукт
Руководитель разработки и внедрения программного продукта	<ul style="list-style-type: none">формирование плановконтроль выполнения плановорганизационная работа (в том числе и с Заказчиком)концептуальная архитектура решениячасть аналитической работы
Руководитель группы	<ul style="list-style-type: none">оценка длительности и трудоемкости задач в процессе планированияконтроль выполнения планов группойраспределение работ внутри группыконцептуальная архитектура решениячасть аналитической работыорганизация сбора требований заказчикасоответствие деятельности группы бизнес-процессу разработкиработа группы с Заказчиком
Аналитик	<ul style="list-style-type: none">сбор требований заказчикаразработка ТП на функциональностьразработка планов тестированияконцептуальное тестирование функциональностиразработка пользовательской документации
Архитектор	<ul style="list-style-type: none">архитектура решения, и соответствие ее требованиям к решениюразработка РП на функциональность (определяет принципиальные моменты, в дальнейшем их детализирует в рамках РП Разработчик)контроль качества кода, и соответствие его проектным решениям по архитектуререпозиторий информации по архитектуре решенияучаствует в формировании планов и оценке сложности и длительности задачучаствует в комплексном тестировании

Программист	<ul style="list-style-type: none"> • разработка РП (при участии Архитектора в процессе выработки принципиальных решений) • разработка функциональности • качество кода • исправление ошибок в коде • проведение первичного тестирования кода • участвует в комплексном тестировании кода
Тестировщик	<ul style="list-style-type: none"> • тестирование функциональности • написание Unit тестов • участвует в разработке планов тестирования
Билд-инженер	<ul style="list-style-type: none"> • сборка версии • выпуск версии (после тестирования) • подготовка сопроводительных документов к версии
Инженер службы внедрения	<ul style="list-style-type: none"> • инсталляция программного продукта • обновление версии программного продукта • анализ сбоев работы программного продукта
Специалист службы сопровождения	<ul style="list-style-type: none"> • администрирование программного продукта • администрирование баз данных • системное администрирование • выявление сбоев работы программного продукта • анализ причин сбоев (в том числе с привлечением Инженера службы внедрения)

При реализации конкретных проектов допускается дополнение состава участников (ролей), predetermined настоящим регламентом, а также выполнение нескольких ролей одним исполнителем.

Этапы жизненного цикла

Жизненный цикл программного продукта состоит из ряда этапов, на которых программный продукт планируется, создается, внедряется, эксплуатируется, сопровождается и списывается. Этапы жизненного цикла объединяются в последовательности, которые могут накладываться друг на друга и/или повторяться в соответствии с границами, величиной, сложностью, меняющимися потребностями и возможностями воплощения программного продукта. Они являются итерационными и реализуются пошаговыми действиями.

Настоящий регламент определяет следующие этапы жизненного цикла программного продукта:

- разработка программного продукта;
- внедрение программного продукта.

Каждым этапом достигается отчетливая цель и вносится вклад в жизненный цикл программного продукта в целом. Каждый этап учитывается при планировании и выполнении жизненного цикла программного продукта.

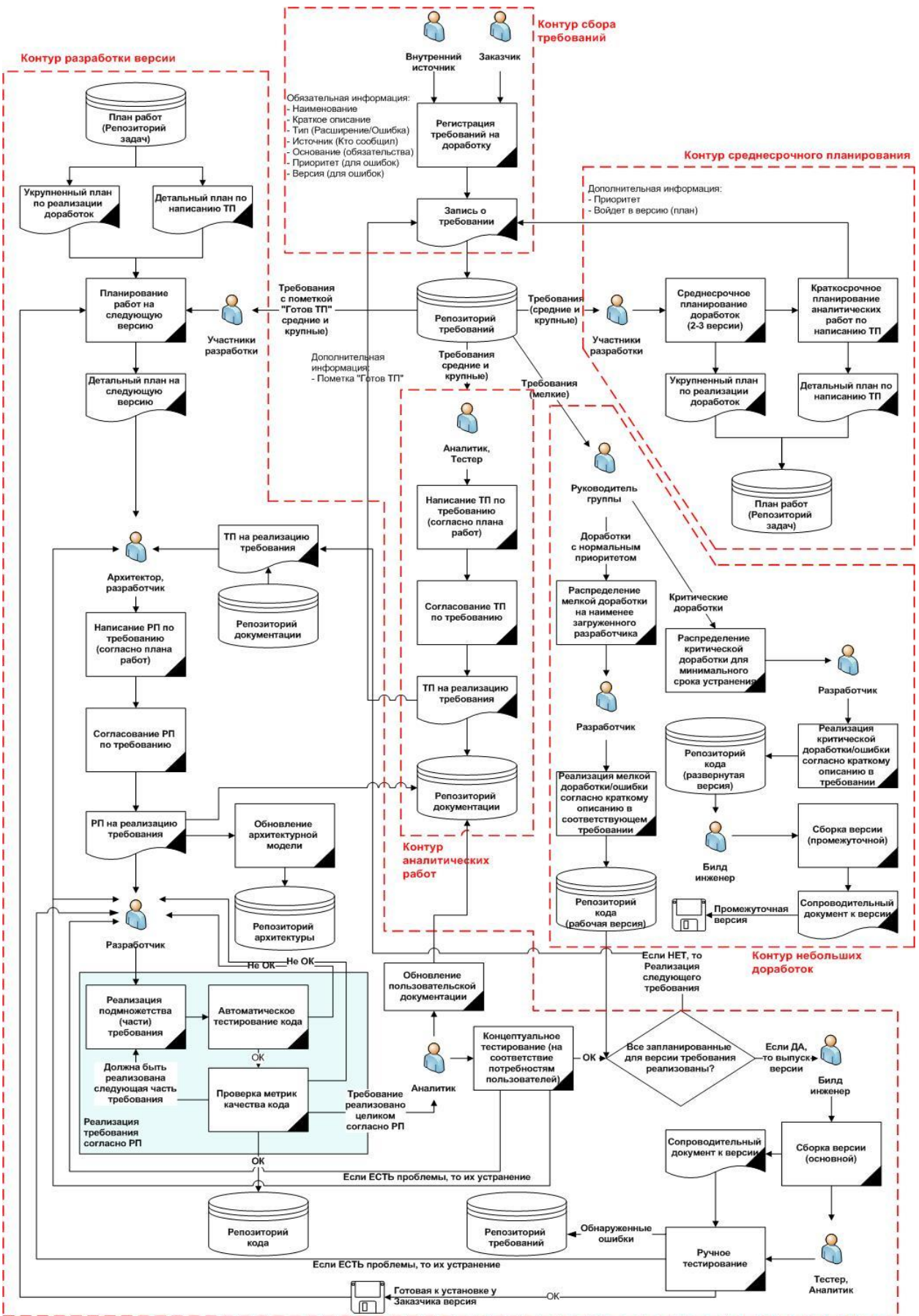
Разработка программного продукта

Настоящим регламентом закрепляется использование итерационного метода разработки в качестве основного метода разработки программного продукта.

Итерационный метод разработки - это метод, при котором определение требований, анализ, проектирование, программирование, интеграция и проверка реализуются постепенно в ходе разработки программного продукта. Итерационный метод подразумевает тесное сотрудничество Заказчика и Разработчика во время создания программного продукта. Итерационный метод характеризуется быстрым вводом в эксплуатацию программного продукта с ограниченными (основными) возможностями, которые определяет сам заказчик, с постепенным добавлением остальных функций, не прерывая эксплуатацию программного продукта.

Схема процесса при итерационном методе разработки приведена ниже. Выделяются следующие группы (контуры) работ:

- контур сбора требований;
- контур среднесрочного планирования;
- контур аналитических работ;
- контур разработки версии (итерация);
- контур небольших доработок.



Контур сбора требований

Основной операцией в данном случае является процедура регистрации требований на доработку. Под требованием понимается в данном случае любое формально описанное условие, которому должен удовлетворять программный продукт.

В частности, требованиями являются:

- ошибки, выявленные при внутреннем или внешнем тестировании, на этапе эксплуатации программного продукта;
- функциональные требования (доработки);
- нефункциональные требования (ограничения, которым должен удовлетворять программный продукт).

Требования могут поступать из различных источников, например:

- представителей Заказчика;
- пользователей программного продукта;
- тестировщиков в составе команды разработки;
- технических заданий при внедрении программного продукта;
- аналитических материалов решений, представленных на рынке.

Все требования хранятся в Репозитории требований. Зарегистрировать требование может любой участник команды разработки, заполнив необходимый набор полей. В дальнейшем все активности по разработке связываются с конкретными требованиями, на реализацию которых они направлены.

После регистрации требования выясняются основания для его реализации (технические задания, договора, аналитика рынка и т.д.). Если оснований не обнаруживается, то требование отвергается или инициируется процесс внесения дополнений в договорные документы. Если основания есть, то требование может быть использовано в других контурах работ. Также требование может быть уточнено, если из его описания не понятно имеет оно основания или нет.

Все требования делятся на две группы с точки зрения подхода к их реализации:

- мелкие доработки (длительность их реализации до 1 человеко-дня И не требуется аналитической проработки);
- средние и крупные доработки (длительность реализации более человеко-дня ИЛИ требуется аналитическая проработка).

Решение о реализации мелких доработок принимается руководителем группы путем назначения требования на соответствующего разработчика (Контур небольших доработок).

Реализация средних и крупных доработок происходит поэтапно:

- планируется реализация доработок и аналитических работ по их проработке (Контур среднесрочного планирования);
- проводятся аналитические работы, по их результатам составляется Технический проект (ТП) на реализацию требования (Контур аналитических работ);
- в рамках детального планирования работ по разработке текущей версии выбираются проработанные требования с готовыми ТП и включаются в состав задач для рабочего проектирования и реализации в текущей версии (Контур разработки версии).

Контур среднесрочного планирования

В рамках контура сбора требований членами команды разработки путем проведения совещания принимается решение о реализации имеющих основания требований. Производится их приоритезация и решается, в какую именно версию должны попасть требования в зависимости от их приоритета.

Распределение требований происходит по нескольким следующим версиям с временным интервалом до полугода. В итоге составляется среднесрочный план, представляющий из себя список требований, которые планируется реализовать в следующих версиях. В карточке каждого запланированного требования помечается эта версия.

На основании среднесрочного плана по реализации требований производится детальное планирование аналитических работ по написанию технического проекта с постановкой задачи по их реализации.

Планы работ хранятся в Репозитории задач.

Контур аналитических работ

Согласно плану работ Аналитик производит аналитическую проработку требования и составляет документ с утвержденной структурой разделов – Технический проект с описанием логики реализации требования.

Технический проект, подготовленный Аналитиком, согласуется с:

- Руководителем разработки и внедрения программного продукта;
- Руководителем группы;
- Архитектором.

После составления и согласования Технического проекта требование помечается как готовое к включению в план разработки версии.

Технический проект, как и остальная документация хранится в Репозитории документации.

Контур разработки версии

Контур разработки версии представляет из себя одну или несколько последовательных итераций разработки: от планирования работ по версии, до ее выпуска. Цель каждой итерации - запустить в эксплуатацию одну или несколько новых протестированных и готовых к использованию функций. После выпуска одной версии, начинаются работы по следующей версии.

При планировании работ по версии команда разработки просматривает требования, выбирая среди них те, которые:

- в соответствии со среднесрочным планом должны быть реализованы в рамках настоящей версии;
- по которым завершена аналитическая проработка.

Также в состав работ по версии могут включаться требования, имеющие критичную важность и наивысший приоритет. В этом случае их аналитическая проработка планируется в рамках работ по версии. Однако это вариант не является основным.

После выделения требований, подлежащих разработке в рамках настоящей версии, составляется детальный план разработки этой версии, включающий в себя все виды работ.

Затем согласно плану Руководитель группы разбивает требования на задачи - программные элементы, реализуемые в минимальные сроки. Задачи распределяются между конкретными программистами. В зависимости от сложности задач время на их разработку может перераспределяться между исполнителями.

По мере завершения задач их код интегрируется в законченную функцию и тестируется разработчиком (при помощи созданных тестов или вручную). При отрицательном результате теста функция вновь разбивается на задачи и проверяется их реализация.

После реализации всех частей одного требования Программист демонстрирует разработанный функционал Аналитику – происходит его концептуальное тестирование, т.е. тестирование на предмет соответствия потребностям пользователей. Если выявляются несоответствия потребностям пользователей, то происходит либо доработка требования и, затем, реализованного функционала либо сразу доработка функционала (в случае мелких замечаний, не требующих изменения требования и архитектурной модели).

В случае успешного концептуального тестирования Программист приступает к реализации следующего требования в соответствии с планом работ по версии.

Также после согласования новой функциональности Аналитик обновляет пользовательскую документацию на решение.

После того, как все доработки, запланированные в версии, реализованы, Билд-инженер производит сборку версии. Билд-инженер, формирует Сопроводительный документ к версии, в котором:

- описываются параметры версии (номер, дата выпуска, перечень файлов и др.);
- перечисляются реализованные в версии требования (новый функционал, устраненные ошибки);
- содержится информация о развертывании версии (последовательность действий);
- если в рамках версии изменилась структура хранилищ данных, то в рамках этого документа прикладываются SQL-скрипты для перехода со старой структуры на новую.

Собранная версия тестируется Тестировщиком при участии Аналитика согласно плану тестирования, являющегося составной частью Технического проекта, разработанного Аналитиком. Выявленные ошибки регистрируются в Репозитории требований в виде требований с типом «Ошибка тестирования версии».

Выявленные ошибки устраняются Разработчиками, которые реализовывали соответствующий функционал, при необходимости модифицируется Рабочий проект и Архитектурная модель.

После того, как ошибки тестирования версии устранены тестирование повторяется. Версия выпускается в следующих случаях:

- все «Ошибки тестирования версии» устранены;
- принимается решение о выпуске версии с рядом не устраненных ошибок.

После того как принято решение о выпуске версии Билд-инженер готовит дистрибутивы и Сопроводительный документ (актуализирует его при необходимости) и передает его для развертывания.

После выпуска версии работы в рамках данной итерации считаются завершенными и начинается работа над следующей версией.

Контур небольших доработок

Небольшие доработки – доработки с длительностью реализации менее человеко-дня, не требующие написания Технического проекта.

Все небольшие доработки делятся на две группы:

- с нормальным приоритетом – включаются в следующую основную версию;

- с критическим приоритетом – для них выпускается промежуточная версия путем внесения требуемых изменений в экземпляр с кодом текущей развернутой версии.

Решение о реализации мелкой доработки с нормальным приоритетом принимается Руководителем группы путем анализа загрузки разработчиков по группе. Наименее загруженные на реализации плановых доработок по версии Разработчики занимаются устранением мелких доработок с нормальным приоритетом.

После устранения мелких доработок с нормальным приоритетом они попадают в рабочую базу кода и выпускаются со следующей основной версией.

Решение о реализации мелкой доработки с критическим приоритетом принимается Руководителем разработки и внедрения программного продукта совместно с Руководителем группы. В этом случае Программист, наиболее хорошо знакомый с возникшей проблемой может быть временно переключен с плановых работ на ее устранение.

После устранения критических мелких доработок (одной или нескольких сразу) инициируется процедура выпуска промежуточной версии. Сборку осуществляет Билд-инженер, также он готовит Сопроводительный документ в котором указывает какой номер данной промежуточной версии и перечень критических мелких доработок устраненных в ней.

После выпуска промежуточной версии Программисты вносят изменения, соответствующие критическим мелким доработкам, в основную рабочую версию.

Информационная модель процесса

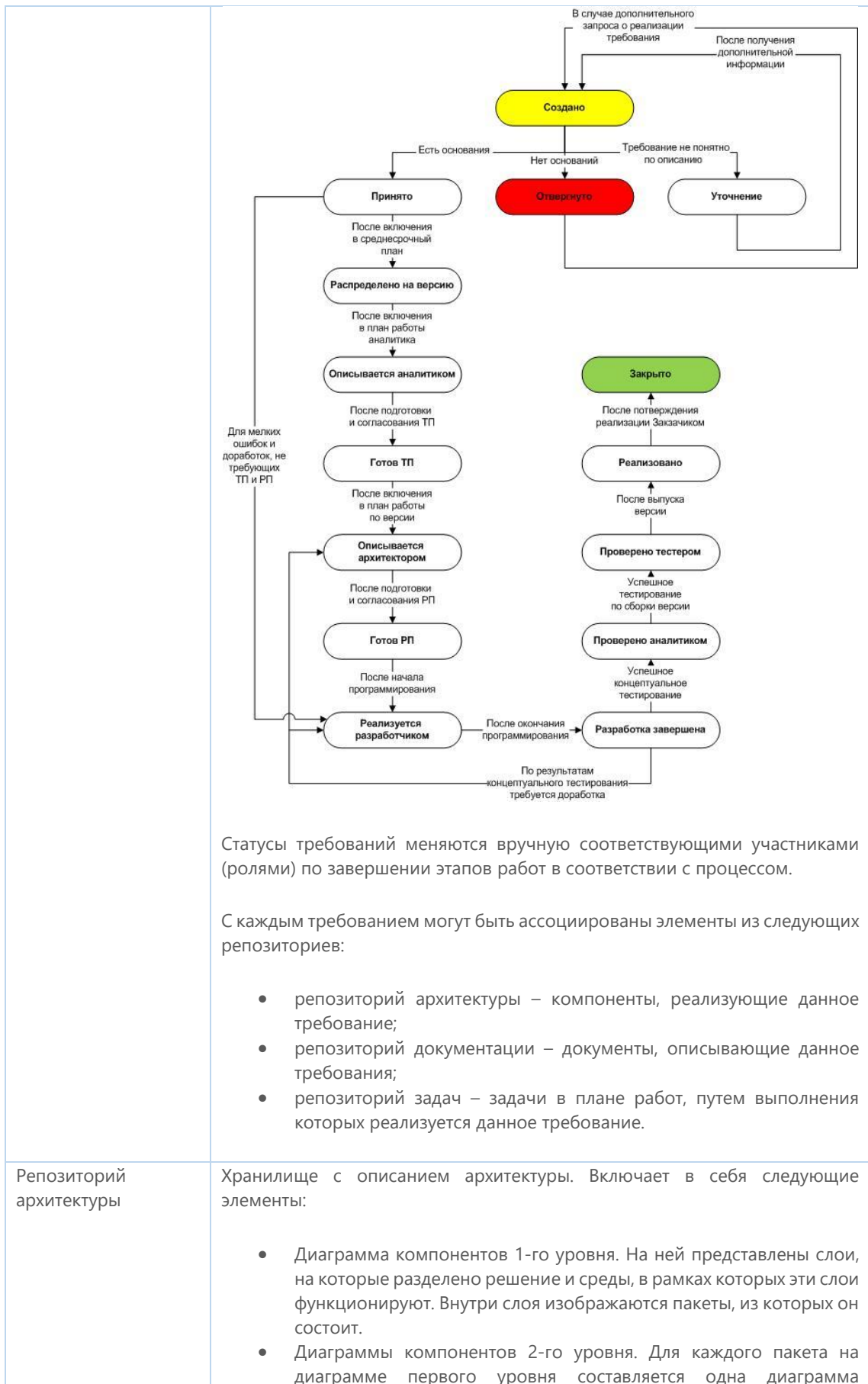
Все виды информации в рамках разработки программного продукта распределяются по пяти хранилищам (репозиториям):

- репозиторий требований;
- репозиторий архитектуры;
- репозиторий документации;
- репозиторий кода;
- репозиторий задач (план работ).

Элементы в одном хранилище могут быть связаны с элементами в другом хранилище. Например, требования в репозитории требований могут быть связаны с компонентами из репозитория архитектуры, которыми они реализуются.

В таблице ниже дается краткое пояснение по содержимому репозитория и описаны взаимосвязи между их элементами.

Репозиторий	Краткое описание
Репозиторий требований	<p>Иерархическая структура групп, содержащая требования к решению, являющиеся исходными основаниями для каких-либо работ по разработке:</p> <ul style="list-style-type: none"> • функциональные требования (функции решения); • нефункциональные требования; • ошибки (баг-трекинг). <p>Группы могут быть вложенными. Требования вложенными быть не могут.</p> <p>Каждое требование содержит следующую информацию:</p> <ul style="list-style-type: none"> • наименование; • краткое описание; • текущий статус (отражает текущий этап жизненного цикла требования); • тип требования (Расширение / Ошибка); • основание (ссылки на договорные документы, ТЗ и иные обязательства); • приоритет; • версия (в которой планируется реализовать требование). <p>Поле «Текущий статус», отражающее жизненный цикл требования, может иметь следующие значения:</p>



	<p>компонентов с перечнем компонентов, входящих в состав этого пакета.</p> <ul style="list-style-type: none"> • Описание компонентов. Для каждого компонента на диаграмме 2-го уровня готовится документ с описанием его интерфейсов и внутренней логики реализации. • Справочник классов. Автоматически генерируемый по коду навигатор по классом, реализованным в системе. <p>С каждым пакетом или компонентом на моделях 2-го уровня в репозитории архитектуры могут быть ассоциированы элементы из следующих репозиторий:</p> <ul style="list-style-type: none"> • репозиторий требований – требования, при реализации которого используется данный компонент; • репозиторий документации – общие описания архитектуры, описания слоев, компонентов (интерфейсные и внутренние части), классов; • репозиторий кода – ссылки на конкретные программные модули, реализующие данный компонент.
Репозиторий документации	Хранилище файлов различных типов, используемых как сами по себе (например, Договорные документы, Протоколы, Акты и др.), так и в связке с элементами из репозитория требований (ТП, РП) и репозитория архитектуры (описания компонентов, слоев).
Репозиторий кода	<p>Весь код решения с поддержкой ветвления версий.</p> <p>Для целей срочной реализации критических доработок в рамках Репозитория кода существуют два экземпляра кода решения:</p> <ul style="list-style-type: none"> • рабочая версия – в ней производятся все доработки согласно плана работ по версии и мелкие доработки нормального приоритета; • текущая развернутая версия – эксплуатируемая в настоящий момент версия – в ней реализуются критические мелкие доработки и выпускается промежуточная версия каждый раз, когда возникает необходимость устранить какие-то принципиальные моменты не дожидаясь выпуска следующей версии. <p>После выпуска промежуточной версии, вносятся соответствующие изменения в Рабочую версию в полуавтоматическом режиме.</p>
Репозиторий задач (план работ)	План работ, содержащий задачи для всех участников проекта. Задачи связаны с требованиями, на реализацию которых они направлены. Также задачи связаны с CheckIn`ами кода, который создавался или модифицировался в рамках выполнения задачи.

Внедрение программного продукта

Процесс внедрения программного продукта осуществляется в рамках проектной деятельности и используются для определения необходимости в программном продукте, преобразования этой потребности в эффективный программный продукт, использования программного продукта с целью предоставления необходимых услуг, поддержки предоставления этих услуг и, когда программный продукт снимается со службы, утилизации этого продукта.

Внедрение программного продукта включает в себя следующие группы работ:

1. предпроектные работы;
2. разработка;
3. внедрение;
4. эксплуатация;
5. сопровождение;
6. утилизация.

Предпроектные работы

Предпроектные работы начинаются с первоначального осознания Заказчиком потребности создания новой или модификации существующей системы информационной безопасности организации. Данный этап является периодом начального изучения, сбора фактов и анализа, изучения действующего законодательства и существующей организационной структуры, обзора существующих решений.

Посредством анализа, определения осуществимости, оценки затрат, маркетинга, интеллектуальных способностей и материально-технического снабжения, изучения альтернатив, а также экспериментальной разработки, вырабатываются решение для удовлетворения потребности или реализации концепции. Также на этом этапе определяется потребность в одной или нескольких подсистемах для реализации полноценной системы информационной безопасности.

На этом этапе принимается решение по использованию программного продукта в организации. В случае принятия положительного решения лица, заинтересованные в создании новой или модификации существующей системы информационной безопасности организации, обращаются к руководителю разработки и внедрения программного продукта со служебной запиской, содержащей, как минимум, следующую информацию:

- ФИО;
- контактная информация для обратной связи;
- наименование организации;
- пояснение цели запроса;

- срочность реализации;
- деятельность (процессы), подлежащие автоматизации;
- общего видения (модели) системы информационной безопасности (или группы взаимосвязанных систем), выполняемых ею функций, описания информационного пространства и взаимодействия с другими внешними системами;
- другую информацию.

В случае, если обращение формируется с целью принятия решения по использованию программного продукта в организации, при необходимости Разработчик программного продукта может провести демонстрацию возможностей программного продукта, обследование области деятельности, в которой будет функционировать программный продукт, анализ уровня информатизации данной области, выяснить необходимость модификации программного продукта. На основе полученных знаний Разработчик формирует бизнес-предложение.

Если бизнес-предложение удовлетворяет Заказчика, принимается решение о запуске Проекта, заключении договора, назначении руководителя проекта и утверждении плана Проекта.

План проекта должен содержать, как минимум, следующую информацию:

- заинтересованные стороны, взаимодействующие организации и сторонние системы;
- определяются роли, ответственность и полномочия сторон;
- цели, назначение и основная функциональность программной системы;
- перечень работ;
- календарный план реализации проекта и другие необходимые проектные планы.

Руководитель проекта выполняет следующие функции:

- анализ возможностей и способов реализации Проекта;
- организация процессов, необходимых для реализации Проекта;
- координация действий участников Проекта;
- проведение консультаций с участниками внедрения для решения спорных вопросов;
- назначение исполнителей работ по Проекту;
- тестирование.

Разработка

Разработка начинается с достаточно подробного уточнения требований Заказчика, которые анализируются и преобразуются в приемлемые требования к программному продукту. Требования Заказчика должны включать потребности и

желания всех сторон, заинтересованных в жизненном цикле программного продукта. Требования Заказчика выражают предполагаемое взаимодействие программного продукта с заинтересованными лицами и являются эталоном, по которому проверяется функциональность программного продукта и его способность предоставлять услуги.

На основании анализа требований заинтересованного лица Разработчик с участием Заказчика разрабатывает техническое задание.

В техническом задании Разработчик обязан установить оговоренные требования к программному продукту, включая спецификации качественных характеристик.

Техническое задание согласовывается с руководителями организаций (подразделений), участвовавших в его разработке, и утверждается Заказчиком.

Заказчик обязан принять меры к адаптации инфраструктуры пользователя и приспособить его деятельность к программному продукту, т.е. принять решение относительно бизнес-процессов, подлежащих автоматизации и бизнес-ролей, участвующих в процессе. Эти действия также могут включать адаптацию штатной структуры, разработку организационно-распорядительной документации, организацию необходимых помещений, приобретение оборудования, организацию учебы сотрудников и т.д.

На основании требований технического задания Разработчик формирует техническое решение, определяющее полный набор технически и коммерчески жизнеспособных системных элементов, из которых конфигурируется программный продукт. Решения этого процесса оформляются в техническом проекте. Технический проект необходим для трансформации требований технического задания в описание реализации конкретных модулей или элементов программного продукта.

Технический проект согласовывается с Руководителем разработки и внедрения программного продукта.

На основании результатов данного процесса определяется стратегия адаптации и тестирования программного продукта, детализирующая план разработки.

По завершении адаптации и тестирования программного продукта осуществляется сборка программной системы в соответствии со структурой проекта. Этим процессом программные элементы собираются в работоспособную систему с частичной или полной конфигурацией с целью создания программной системы, предусмотренной требованиями Заказчика. Созданная конфигурация регистрируется и предоставляется заказчику для дальнейшей проверки и приемки.

Цель процесса проверки заключается в том, чтобы продемонстрировать, что характеристики и функциональность программного продукта соответствуют

оговоренным требованиям технического задания. Этот процесс предоставляет аналитическую информацию, необходимую для проведения корректирующих действий. Такие действия направлены на корректировку несоответствий в реализованной программной системе/программном элементе или в процессах, воздействующих на них.

Процесс проверки заключается в планировании и проведении квалификационного тестирования, за действия которого отвечает Разработчик.

Во время квалификационного тестирования проводятся следующие проверки:

- функциональное тестирование программного продукта на соответствие требованиям технического задания;
- нагрузочное тестирование, имитирующее нагрузки на программный продукт больше пиковых значений процесса эксплуатации;
- проверки граничных значений входных данных, при которых программная система меняет свое поведение;
- удобство использования программного продукта и дружелюбность пользовательского интерфейса;
- анализ примененных технологий, алгоритмов и программного кода;
- проверка документации на наличие, содержание и качество исполнения, а также соответствия требованиям руководящих документов.

По итогам квалификационного тестирования составляется протокол тестирования.

Ошибки и замечания, выявленные во время квалификационного тестирования, анализируются Разработчиком совместно с Заказчиком и сортируются по трем группам:

- критические ошибки - ошибки, повлекшие остановку технологического процесса или сбой в программе;
- умеренные ошибки - ошибки, вызывающие неудобство в работе или оказывающие отрицательное влияние на производительность или безопасность системы, а также ограничивающие функциональность системы (без прерывания технологического процесса) в рамках технического задания;
- ошибки процесса проектирования - ошибки или замечания, требующие расширения или изменения функциональности программного продукта, выходящие за пределы технического задания.

После анализа проблем возможны следующие решения:

- ошибки первой группы устраняются до передачи в опытную эксплуатацию, для чего выделяется дополнительное время и корректируется календарный план. После устранения ошибок первой группы программный продукт проходит повторное квалификационное тестирование;
- ошибки второй группы устраняются во время опытной эксплуатации;

- замечания и ошибки третьей группы анализируются Разработчиком совместно с Заказчиком. По каждому пункту этой группы принимается отдельное решение, которое оформляется совместным протоколом заседания. На основании совместного решения заключается новый договор (контракт) или дополняется существующий. Модификации программного продукта в рамках нового договора (дополнения) должны проходить полный жизненный цикл согласно настоящему регламенту.

В случае отсутствия критических ошибок и замечаний, не позволяющих начать процесс внедрения, Разработчик передает программную систему Заказчику для проведения опытной эксплуатации. При этом составляется акт передачи в опытную эксплуатацию.

Внедрение

Внедрение начинается с процесса утверждения. Цель процесса утверждения заключается в предоставлении объективного доказательства, что услуги, предоставляемые программным продуктом или любым программным элементом, удовлетворяют утвержденным требованиям технического задания и потребностям заинтересованных лиц.

Действия процесса утверждения обычно инициируют проведение опытной эксплуатации. Опытная эксплуатация программного продукта предоставляет информацию о его поведении в реальной обстановке эксплуатации с использованием реальных данных. Опытная эксплуатация необходима для ознакомления пользователя с внедряемым программным продуктом и выявления отклонений в его работе.

В процессе утверждения выполняется сравнительная оценка и подтверждается, что были правильно реализованы требования заинтересованных лиц к программному продукту. При выявлении отклонений они регистрируются и осуществляются корректирующие действия.

Все отклонения в работе и замечания (в рамках технического задания), выявленные во время опытной эксплуатации, устраняются разработчиком до окончания опытной эксплуатации. В случае необходимости корректируется календарный план.

Утверждение программного продукта подтверждается заинтересованными лицами путем подписания акта о завершении разработки. Утверждение акта должно производиться сторонами согласно договорным процессам.

На основании акта о завершении разработки начинается деятельность по развертыванию программного продукта в местах его функционирования. На этом этапе осуществляются подготовка инфраструктуры пользователя и оборудование помещений, в которых будет осуществляться функционирование программного продукта, проверка работоспособности программного продукта в реальных

условиях эксплуатации, а также организуется обучение персонала работе с программным продуктом или его компонентами.

Возможно поэтапное развертывание программного продукта. Например, в ограниченных местах его функционирования для проведения опытной эксплуатации или по мере внедрения программного продукта в территориальных подразделениях.

В случае, если внедряемый программный продукт заменяет уже существующую систему, во время процесса перехода возможна одновременная работа существующей и внедряемой программных систем. В этом случае необходимо обеспечить переход на новый программный продукт и, по требованию Заказчика, согласно договору (плану работ) осуществить перенос данных, при этом обеспечивая их целостность.

Процесс перехода завершается выполнением Разработчиком договорных обязательств, но развертывание может продолжаться все время эксплуатации программного продукта. Завершение процесса перехода подтверждается актом передачи в эксплуатацию программного продукта. Утверждение акта должно происходить согласно договорным обязательствам.

Основная часть внедрения заканчивается вводом программного продукта в эксплуатацию.

Эксплуатация

Эксплуатация начинается с установки и документального подтверждения начала использования программного продукта. Использование программного продукта производится в целях получения требуемых программных услуг при непрерывной функциональной и стоимостной эффективности. Этот этап продолжается до начала этапа утилизации.

Начало эксплуатации устанавливается Заказчиком на основании акта ввода в эксплуатацию. За действия в рамках данного процесса отвечает Заказчик.

Заказчик управляет процессом эксплуатации на уровне проекта, создает инфраструктуру этого процесса, приспособливает ее к требованиям проекта.

Процессом эксплуатации Заказчик контролирует получение услуг от программного продукта, оценивает взаимодействие непосредственных пользователей и программного продукта, а также оказывает консультации непосредственным пользователям и осуществляет подготовку кадров. Для поддержания процесса получения программных услуг проблемы, выявленные в ходе эксплуатации, должны регистрироваться и анализироваться. После анализа пользователь препровождает запросы непосредственного пользователя к сопроводителю программного продукта. Предложения и запросы оформляются в виде

предложения о модификации или сообщения о проблеме. Все запросы должны быть прослежены вплоть до их завершения. Взаимодействие пользователя и сопровождаителя должно быть определено процессами договора.

Действия (либо отсутствие каких-либо предпринимаемых действий) по выявленным проблемам могут быть следующими:

- обслуживание программного продукта;
- инициирование действий сопровождения при незначительных модификациях;
- дополнительное инициирование разработки в случае крупных модификаций;
- инициирование списания.

Сопровождение

Сопровождение программного продукта заключается в поддержании его способности предоставлять услуги, а также модификации программного продукта при сохранении его целостности. Этим процессом контролируется работа программного продукта, регистрируются проблемы для анализа в журнале регистрации обращений, предпринимаются предупреждающие и корректирующие действия, а также действия по адаптации и усовершенствованию программного продукта.

Данный процесс включает следующие действия:

- подготовка процесса сопровождения;
- анализ проблем и изменений;
- внесение изменений;
- проверка и приемка при сопровождении;
- переход (перенос).

Применяются следующие основные типы сопровождения:

- корректирующее сопровождение;
- предупреждающее сопровождение;
- адаптивное сопровождение;
- полное сопровождение.

Планирование сопровождения начинается на предыдущих этапах. Выходными элементами этапа сопровождения являются следующие основные артефакты:

- концепция сопровождения;
- план сопровождения;
- согласованный вариант изменений;
- измененный программный продукт и документация;
- акт передачи в эксплуатацию модифицированного программного продукта.

Управляет процессом сопровождения сопроводитель программного продукта на уровне проекта. Он должен организовать управление процессом, в рамках договорных отношений создать инфраструктуру процесса и приспособить процесс к требованиям конкретного проекта.

Данный этап завершается одновременно с завершением эксплуатации. При необходимости на этапе сопровождения проводятся обучение и переподготовка персонала.

Утилизация

Утилизация предусматривает изъятие из эксплуатации программного продукта и связанных с ним программных подсистем и вспомогательных программных процессов. Изъятие программного продукта из эксплуатации также может быть инициировано окончанием срока лицензии на эксплуатацию данного продукта или по просьбе Заказчика. За действия процесса изъятия отвечает сопроводитель.

Сопроводитель, выполняющий изъятие программного продукта из эксплуатации, должен решить следующие задачи: разработать календарный план изъятия из эксплуатации, уведомить пользователей и всех заинтересованных субъектов об изъятии программного продукта из эксплуатации и архивировать соответствующие данные.

Заказчик определяет способ и место хранения выведенных из эксплуатации программных элементов и данных. Процесс изъятия завершается составлением акта о списании программного продукта.

Планирование этапа утилизации начинается на предыдущих этапах.

В результате успешной реализации утилизации:

- разработан календарный план изъятия из эксплуатации;
- заинтересованным сторонам направлены уведомления о намерениях по изъятию из эксплуатации;
- уничтожены или помещены на хранение программные элементы;
- сохранена информация, полученная в результате создания и эксплуатации системы;
- факт изъятия подтвержден актом о списании.

Деятельность по внутреннему тестированию должна быть непрерывной. Тесты для проверки программных элементов должны создаваться сразу, как только эти элементы реализованы в коде. Первоначально написанный тест должен использоваться многократно.

Результатом выполнения тестов должен быть список ошибок, который заносится в протокол тестирования.

Все тесты, созданные в процессе реализации очередной версии, сохраняются и применяются при тестировании следующих версий.