

ООО «Джет Софт»

Программа для ЭВМ

Информационно-аналитическая система Jet BI

**Описание процессов, обеспечивающих поддержание
жизненного цикла ПО**

**Москва
2020**

Содержание

1	Общая информация.....	3
2	Процессы, обеспечивающие поддержание жизненного цикла Jet BI.....	4
3	Стадии создания ПО Jet BI.....	5
3.1	Инициация проекта	5
3.2	Исполнение проекта.....	5
3.3	Управление требованиями.....	6
3.4	Резервирование версий	6
3.5	Финальная фаза проекта	6
4	Общее описание процессов поддержания жизненного цикла ПО	7
5	Стадии проекта	8
5.1	Инициация проекта (начальная фаза).....	8
5.2	Исполнение проекта.....	11
5.2.1	Исполнение проектных задач Jet BI.....	15
5.2.2	Управление требованиями.....	15
5.2.3	Процесс контроля версий.....	16
5.2.4	Политика релизов	16
5.2.5	Тестирование ПО	20
5.2.6	Процесс управления дефектами	21
5.3	Финальная фаза проекта	23

1 Общая информация

Здесь и далее по тексту документа используются следующие определения:

Компания – разработчик информационно-аналитической системы Jet BI, компания ООО «Джет Софт».

Заказчик – организация-потребитель сервисных функциональных возможностей, обеспечиваемых ИАС Jet BI. В случае внутренней разработки (внутреннего заказа) заказчиком разработки является юридическое лицо – компания ООО «Джет Софт».

Исполнитель – компания, выполняющая требования Заказчика в части сбора требований, разработки, тестирования и обеспечения жизненного цикла ПО. В контексте данного документа – это подразделения, непосредственно задействованные при разработки ИАС Jet BI в компании ООО «Джет Софт».

2 Процессы, обеспечивающие поддержание жизненного цикла ПО Jet BI

При разработке ПО использована принятая в Компании методология разработки информационных систем Rational Unified Process (RUP), определяющая итерационный процесс разработки программного обеспечения.

Подробное описание принятых в Компании процессов разработки программного обеспечения и поддержания жизненного цикла ПО приведено в разделе 4 «Общее описание процессов поддержания жизненного цикла ПО» и разделе 5 «Стадии проекта».

3 Стадии создания ПО Jet BI

В соответствии с методологией RUP создание программного обеспечения ИАС Jet BI (далее – ПО Jet BI) и разработку функциональности можно разделить на три стадии:

- Инициация;
- Исполнение;
- Завершение.

3.1 Инициация проекта

Основой для выполнения работ по настоящему проекту послужили возникшие потребности производственных подразделений Заказчика. После получения исходных требований были выполнены следующие работы:

- сбор всей необходимой информации;
- оценка объёма работ;
- формирование плана разработки, тестирования и внедрения Системы;
- оформление результатов Стадии «Инициализация проекта» – подготовка следующих документов:
 - описание функциональных требований;
 - план реализации со сроками;
 - оценка трудозатрат.

3.2 Исполнение проекта

Предварительные функциональные требования, сформулированные на стадии «Инициализация проекта», использовались в качестве исходных данных для последующего анализа.

Проектные работы включили в себя:

- детализацию и уточнение функциональных и нефункциональных требований;
- разработку технических требований с учётом детализации и уточнения исходной информации;
- постановку технических задач непосредственным исполнителям – разработчикам ПО;
- тестирование разрабатываемой функциональности;
- документирование ПО.

3.2.1 Управление требованиями

Все поступившие от Заказчика требования задокументированы, проанализированы и реализованы в программном обеспечении Jet BI.

3.2.2 Резервирование версий

Выдаваемая Заказчику версия программного обеспечения Jet BI, включая исходные коды и документацию, сохранена в отдельном хранилище данных. В хранилище также сохранены архивные записи ПО. Архивирование проводилось с целью освобождения места для хранения новых записей.

3.3 Стадия завершения проекта

Финальная стадия заключалась в сдаче ПО Заказчику и его передаче его на гарантийное обслуживание.

На находящееся в эксплуатации программное обеспечение Jet BI распространяются гарантийные обязательства Исполнителя (см. п. 5.3).

4 Общее описание процессов поддержания жизненного цикла ПО

При реализации проектов в Компании используются общепринятые методологии разработки ПО, прежде всего Rational Unified Process (RUP) – итерационный процесс разработки (см. п. 2).

Целью итераций является последовательное осмысление стоящих проблем, наращивание эффективности решений и снижение риска потенциальных ошибок. По окончании каждой итерации создается законченная версия работающего программного продукта.

При таком подходе можно гибко учитывать новые требования или проводить тактические изменения в бизнес-целях. Это позволяет на самых ранних этапах разработки выявлять и разрешать проблемы и, тем самым, снижать затраты.

Rational Unified Process предполагает разработку, реализацию и тестирование архитектуры ПО на самых ранних стадиях выполнения проекта, тем самым устраняя самые опасные риски, связанные с архитектурой. Благодаря этому удастся избежать существенных переработок на последних стадиях, если вдруг выяснится, что выбранное решение не обеспечивает выполнение каких-либо требований.

Разработка информационных систем в Компании ведется с использованием современных технологий и подходов, предназначенных как для быстрой реализации прототипов, так и для создания больших многоуровневых компонентных и масштабируемых решений корпоративного уровня. В своих проектах Компания успешно применяет методологию создания прикладных программных систем, разработанную на базе RUP.

При создании ПО применяются инструментальные средства и технологии таких известных компаний как Postgres Pro, а также собственные оригинальные разработки и технологии Компании.

При разработке ИАС Jet BI и её компонентов применяются технологии C++ и Node.JS. Использование собственных разработок Компании позволяет не начинать реализацию проекта «с нуля» так как имеется банк шаблонов стандартных наработок. Это дает возможность Заказчику получить реально работающую и экономичную систему в максимально короткие сроки.

5 Стадии проекта

Укрупнённо проект по созданию программного обеспечения можно разделить на три стадии:

- Инициация проекта;
- Исполнение проекта;
- Завершение проекта.

5.1 Инициация проекта (начальная фаза)

Инициализация предпроектного обследования выполняется по запросу Заказчика на предоставление технико-коммерческого предложения и проводится на основе информации, которая содержится в запросе. При необходимости Исполнитель проводит одно-два интервью с экспертами Заказчика или запрашивает дополнительные материалы. По итогам предпроектного обследования разрабатывается Коммерческое предложение, в котором представлены следующие артефакты:

- Концепция системы (функциональная и программно-аппаратная архитектура);
- План проекта;
- Оценка трудозатрат;
- Ценовое предложение.

Исполнитель использует в повседневной работе модель расчета трудоемкости и стоимости работ/услуг. Модель базируется на технологии ведения проектов, которая предполагает следующие этапы проекта:

- Техническое задание;
- Проектирование;
- Разработка;
- Тестирование;
- Документирование;
- Управление.

Такая модель расчета трудоемкости и стоимости работ/услуг предназначена для расчета трудоемкости проекта на основании зависимости трудоемкости этапов проекта от трудоемкости этапа разработки и с учетом факторов влияния на различные этапы проекта.

Исходными данными для расчета трудоемкости проекта являются перечень предварительных требований к разрабатываемой системе и верхнеуровневое описание архитектуры ПО.

При расчете трудоемкости проекта выполняется:

- Определение базовой трудоемкости этапа разработки;
- Определение трудоемкости проекта.

Трудоемкость разработки (только для этапа «Разработка») является базой для определения трудоемкости остальных этапов проекта. При расчете базовой трудоемкости учитываются различные аспекты разработки, наличие которых корректирует трудоемкость данного этапа.

Для всех остальных этапов проекта определены нормативы зависимости от трудоемкости этапа «Разработка» (в процентах). Различные этапы могут входить или не входить в расчет трудоемкости того или иного проекта. По каждому этапу определяются факторы влияния, наличие которых увеличивает нормативную трудоемкость данного этапа.

Модель расчета трудоемкости и определяет три типа нормативов:

- нормативы трудоемкости разработки программных компонентов;
- нормативы коэффициентов для аспектов разработки;
- нормативы коэффициентов для этапов разработки и факторов влияния.

5.1.1 Нормативы трудоёмкости разработки

Для расчета нормативов трудоемкости разработки определены следующие типы программных компонентов:

- пользовательский интерфейс (экранные формы, веб-страницы, отчеты, печатные формы и т. д.);
- бизнес-логика (компоненты, реализующие логику работы приложения, алгоритмы обработки данных и т. д.);
- компоненты хранения данных (таблицы БД, серверные объекты, хранимые процедуры и т. д.);
- внешний интерфейс (компоненты, реализующие логику интеграции с внешними системами).

Для каждого типа технических компонентов определены градации сложности (от «Простого» до «Очень сложного») и определены нормы трудозатрат на разработку (в человеко-часах), основанные на экспертной оценке.

Например, при разработке экранных форм классификация компонентов по сложности будет выглядеть так:

- **Простая** – окно авторизации или окно ввода данных с проверкой вводимых данных по формату (без логических связей между различными параметрами).

- **Средняя** – экранная форма, реализующая поиск и фильтрацию данных с постраничным выводом.
- **Сложная** – экранная форма, содержащая не более двух вкладок с проверкой ввода данных по формату и логической взаимосвязи данных.
- **Очень сложная** – экранная форма, содержащая более двух вкладок с проверкой ввода данных по формату и с учетом логических взаимосвязей.

Такая классификация является лишь примерной иллюстрацией, окончательное отнесение компонента к той или иной категории сложности выполняет архитектор.

Норматив трудоемкости разработки технических компонентов определяется в человеко-часах в разрезе сложности компонентов.

5.1.2 Нормативы аспектов разработки

Нормативы коэффициентов для аспектов разработки определяются в процентах относительно нормативов трудоемкости разработки программных компонентов и разработки в целом.

5.1.3 Нормативы для этапов разработки

Нормативы коэффициентов трудоемкости определяются в процентном соотношении к Этапу «Разработка» или другим этапам и факторам влияния.

В Табл. 1 приведен пример нормативов для этапа «Тестирование» и трудоемкости этапа «Управление».

Табл. 1 – Примеры коэффициентов для расчета трудоемкости этапов

Этап	Описание этапа	Детали расчета фактора влияния /этапа
Интеграционное тестирование	Интеграционное тестирование (включает время на исправление найденных во время тестирования критичных дефектов)	Процент от Разработки внешних интерфейсов
Нагрузочное тестирование	Нагрузочное тестирование (включает время на исправление найденных во время тестирования критичных дефектов)	Процент от Разработки
Документирование	Документирование	Процент от Разработки

Фактор влияния – это обстоятельство или условие, увеличивающее трудоемкость этапа или вида работы. Примеры факторов влияния и нормативы соответствующих коэффициентов приведены в Табл. 2.

Табл. 2 – Примеры коэффициентов для расчета факторов влияния

Этап	Описание этапа	Детали расчета фактора влияния /этапа
Техническое задание	Комплексно сложный продукт (необходимо написание ЧТЗ на различные компоненты/модули системы)	Процент от Технического задания

Этап	Описание этапа	Детали расчета фактора влияния /этапа
Проектирование	Использование специфической методологии, нотации, специальные требования Заказчика и т. д	Процент от Проектирования
Разработка	Специфическая среда разработки (например: отсутствие средств рефакторинга и т. д)	Процент от Разработки
Разработка	Обеспечение качества: код-ревью. Этот фактор может не учитываться в случае немасштабного проекта, разработки макета или прототипа.	Процент от Разработки
Разработка	Обеспечение качества: юнит- тестирование. Этот фактор может не учитываться в случае немасштабного проекта, разработки макета или прототипа.	Процент от Разработки
Функциональное тестирование	Необходимость развертывания тестовой среды на площадке Заказчика	Процент от Функционального тестирования
Функциональное тестирование	Необходимость разработки тестовой платформы (автотесты, эмуляторы и т. д). Наличие этого фактора зависит от масштаба, критичности, планов поддержки разрабатываемой системы и т. п.	Процент от Разработки
Интеграционное тестирование	Отсутствие технических специалистов на стороне внешних систем	Процент от Интеграционного тестирования
Документирование	Написание документации на иностранном языке (перевод). За каждый дополнительный язык устанавливается свой норматив. Консультирование переводчика	Процент от Документирования
Документирование	Если Заказчик – Госструктура (требование оформления документации по ГОСТ или использование отраслевых или иных специализированных стандартов)	Процент от Документирования
Документирование	Необходимость подготовки учебных курсов	Процент от Документирования
Документирование	Необходимость сертификации (ФСБ России, ФСТЭК России, Минкомсвязь и т. д.) – полный пакет документов	Процент от Документирования
Управление	Требование ведения работ по разработке продукта на площадке Заказчика	Процент от Управления

5.2 Исполнение проекта

Общая схема технологии разработки ПО представлена на Рис. 1.

Технология разработки ПО

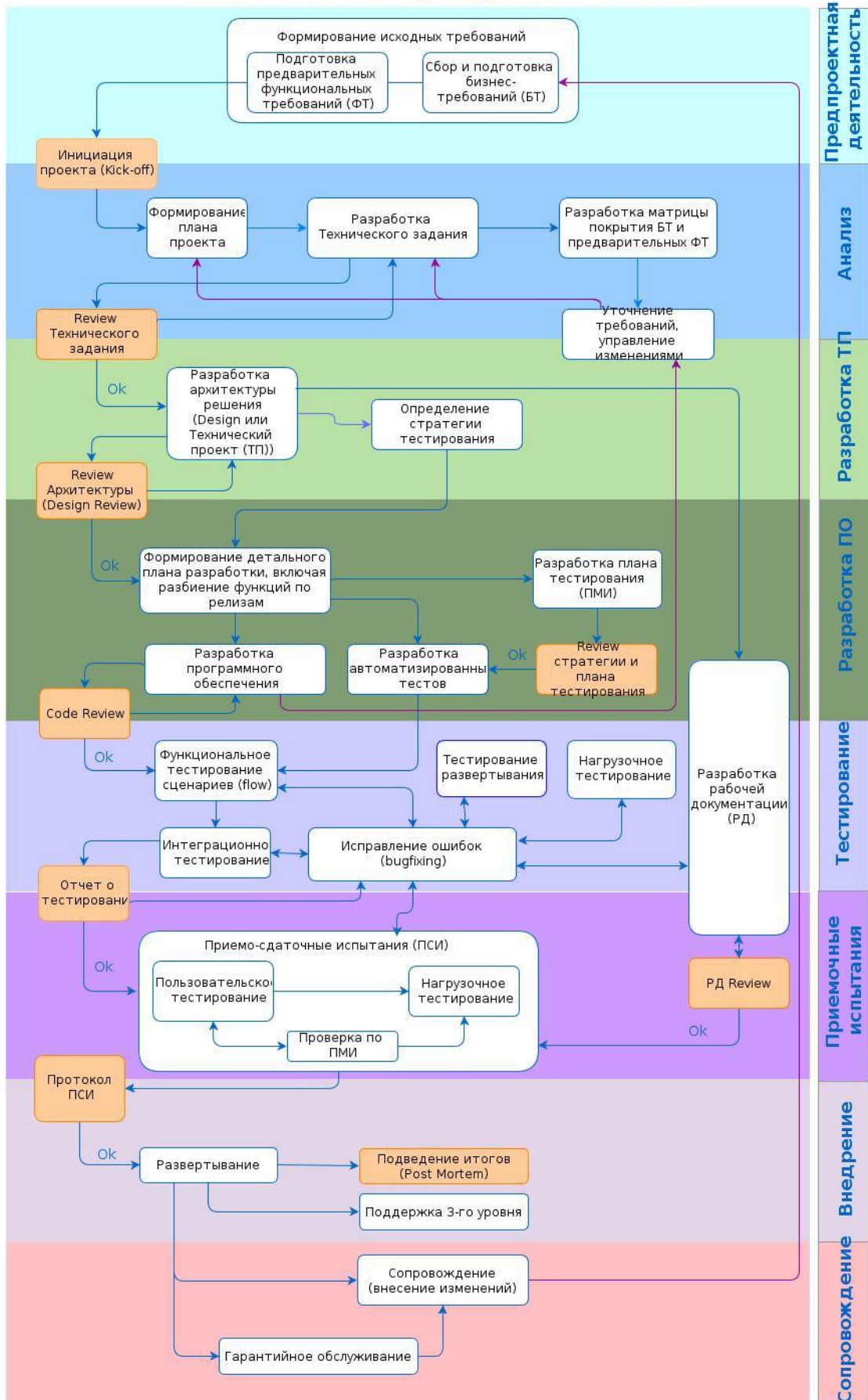


Рис. 1 – Технология разработки программного обеспечения

Разбиение на этапы приведено с учетом времени выполнения той или иной работы и не всегда совпадает с логическим разбиением работ по типам. Например, работы «Определение стратегии тестирования» и «Разработка программы и методики испытаний» относятся к внутреннему тестированию, а на схеме отнесены к этапам «Разработка технического проекта» и «Разработка программного обеспечения» соответственно (см. Рис. 1).

На этапе «Предпроектная деятельность», в начале проекта, формируются бизнес-требования и предварительные функциональные требования, после чего инициируется проект и начинается этап «Анализ». На этапе «Анализ» формируется детальный план проекта, проводится бизнес-анализ, разрабатываются Техническое задание и матрица покрытия. На этапе «Анализ» проводятся следующие работы:

- обследование (проведение серий интервью), детализация и конкретизация бизнес-требований;
- разработка Технического задания;
- описание функциональности системы и бизнес-процессов «как будет». Здесь же определяется логическая структура объектов и их атрибутивный состав.

По итогам этапа должны быть разработаны следующие документы:

- Отчет об обследовании (необязательный документ, необходимость разработки определяется в зависимости от особенностей проекта и согласовывается с Заказчиком)
- Бизнес-требования (как правило, содержатся в запросе на технико-коммерческое предложение, в некоторых случаях требуется их детализация до разработки Технического задания);
- Матрица покрытия бизнес-требований и предварительных функциональных требований (вспомогательный документ);
- Техническое задание (обязательный документ).

Review Технического задания и других результатов этапа «Анализ» происходит как на данном этапе, так и на этапе «Разработка технического проекта».

На этапе «Разработка технического проекта» проводятся следующие работы:

- проектирование (детализация) архитектуры разрабатываемого ПО;
- проектирование физических структур объектов хранения;
- детальное определение интерфейсов взаимодействия с внешними системами;
- проектирование пользовательского интерфейса.

Одновременно определяется стратегия тестирования. Результатом проектирования являются документы и модели, которые дорабатываются и корректируются в процессе реализации и оформляются на этапе «Документирование».

Review Архитектуры происходит и на этапе «Разработка ПО».

Также на этапе разработки ПО формируется детальный план разработки, включая разбиение функций по релизам, и разработка программного обеспечения. Кроме того, на данном этапе разрабатывается план тестирования, который входит и в план тестирования, который как и Code Review (при необходимости) выполняется на этапе «Тестирование». Детализация этапа «Разработка ПО» совпадает с детализацией, определяемой архитектором при расчете трудоемкости.

На этапе «Тестирование» выполняются следующие работы:

- определение стратегии тестирования (на схеме отнесено к этапу «Разработка ТП»);
- разработка программы и методики испытаний (отнесено к этапу «Разработка ПО»); для проектов, не требующих оформления по ГОСТ, разрабатываются сценарии тестирования (в том числе автоматизированного);
- собственно тестирование;
- исправление дефектов по результатам тестирования.

Отдельно, в зависимости от специфики проекта, определяется необходимость нагрузочного тестирования и автоматизированного и (или) регрессионного тестирования. Одновременно, в рамках функционального, интеграционного или нагрузочного тестирования определяется необходимость разработки соответствующих скриптов.

Документирование подразумевает разработку пользовательской и эксплуатационной документации, сопровождающей поставку программного продукта, а также оформление проектной документации, разработанной в ходе проектирования в соответствии со стандартами, используемыми в проекте (ГОСТ, РUP и т. д.). Точный перечень документов, разработка которых требуется при выполнении проекта, определяется в Контракте или Договоре и Техническом задании. В обязательном порядке разрабатываются документы:

- Руководство пользователя;
- Руководство администратора.

Документацию можно разделить на релизную, эксплуатационную и сертификационную. Наиболее сложная и объемная документация – сертификационная.

На этапе «Приемочные испытания» выполняется различные виды тестирования.

5.2.1 Исполнение проектных задач ИАС Jet BI

Информационно-аналитическая система Jet BI (далее – ИАС Jet BI, Система) – предназначена для обеспечения деловой осведомленности и поддержки принятия решений аналитиками, руководителями или другими ответственными работниками коммерческих компаний и государственных структур (далее – организаций).

ИАС Jet BI позволяет решать следующие задачи:

- контролировать проектную и операционную деятельность организации;
- контролировать финансовые и производственные показатели организации;
- строить соответствующие отчеты по аналитической информации о деятельности организации.
- получать сводные отчеты о деятельности организации.

ПО информационно-аналитической системы Jet BI (далее – ПО Jet BI) реализовано в виде единого программного элемента, построенного по технологии клиент-сервер.

ПО Jet BI использует открытые форматы обмена данными. Информационная совместимость со смежными системами обеспечивается на уровнях:

- взаимодействия в пакетном режиме;
- потокового взаимодействия;
- экспорта-импорта документов.

5.2.2 Управление требованиями

При реализации проектов используется систематический подход к процессу управления требованиями. В рамках процесса управления требованиями выполняется: выявление, документирование, верификация и утверждение требований, планирование реализации и отслеживание изменений требований.

Типовой состав участников проектной группы в рамках процесса управления требованиями выглядит следующим образом:

- системный аналитик;
- специалист по требованиям;
- рецензент требований;
- архитектор.

Кроме того, в процессе задействован эксперт предметной области. В большинстве проектов роль эксперта, как правило, играет представитель заказчика, но в ряде случаев эксперт предметной области входит в состав проектной группы со стороны исполнителя.

Процесс управления требованиями изображен на Рис. 2.

5.2.3 Процесс контроля версий

При разработке программного обеспечения Компания придерживается ряда правил и норм, в части применения фиксированной политики релизов, резервирования исходных кодов ПО и использования централизованной системы контроля версий.

5.2.4 Политика релизов

Политика релизов (выпуск версий ПО информационных систем) фиксируется до начала проектных работ и может отличаться в проектах, выполняемых для различных заказчиков. Поэтому, в начале проекта, прежде всего, определяется и согласовывается с заказчиком классификация выдаваемых ему релизов. Например, одной из возможных классификаций является разбиение релизов на Major/Minor-релизы с точки зрения функциональности ПО и Candidate/Final-релизы с точки зрения процесса разработки.

К Major-релизам относятся все релизы с изменениями в функциональности ПО и релизы, содержащие пакеты исправлений выявленных критичных дефектов в функциональности промышленной версии продукта.

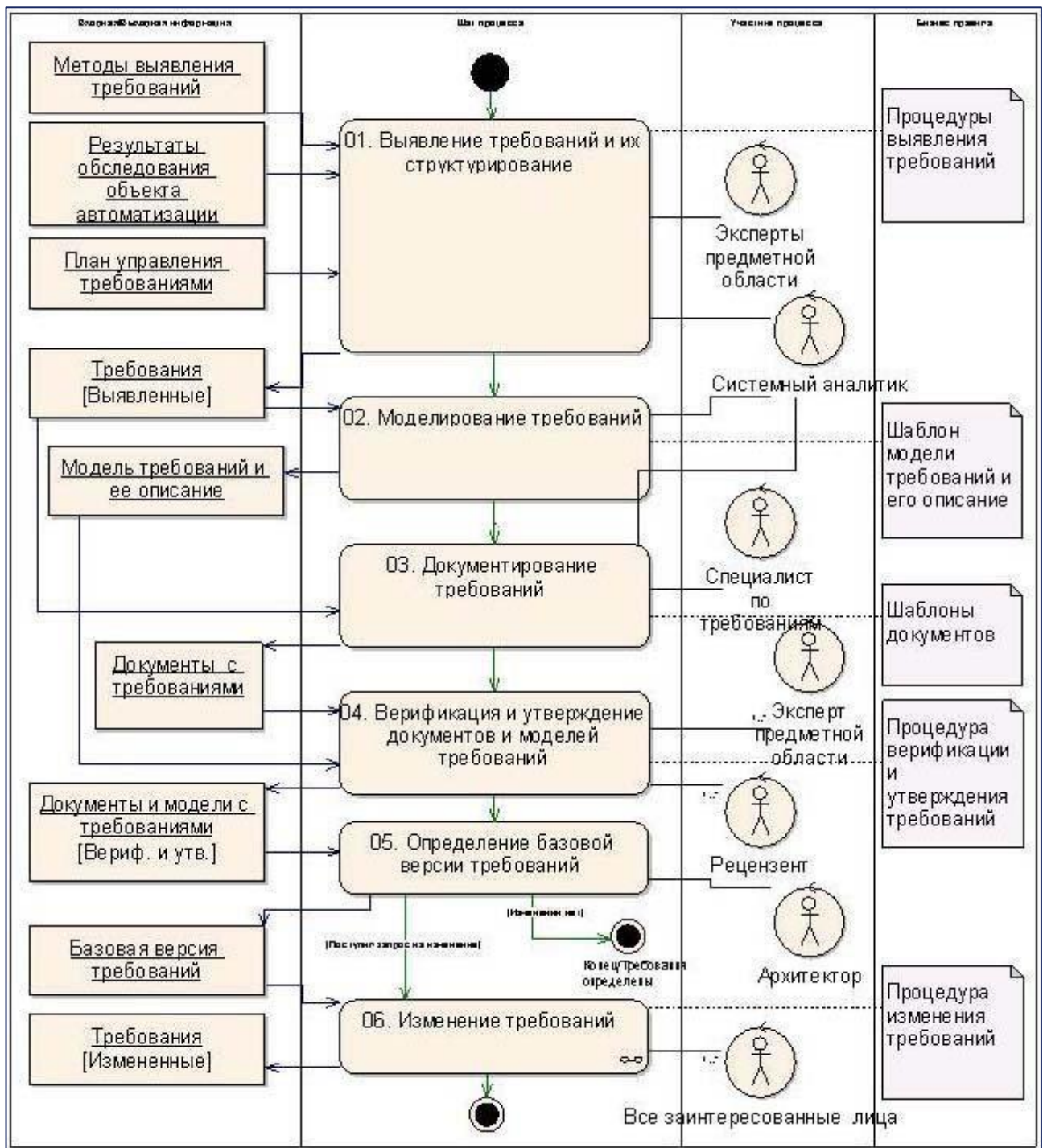


Рис. 2 – Процесс управления требованиями

К Minor-релизам относятся релизы, содержащие пакеты исправлений дефектов, обнаруженных в выданном ранее Major-релизе, не запущенном пока в промышленную эксплуатацию.

К Candidate-релизу относится готовая сборка всех компонентов продукта, содержащая реализованные функциональные требования, предъявляемые к данному релизу, но не прошедшая этап тестирования.

После успешного прохождения этапа тестирования релиза и выдачи его заказчику Candidate-релиз переходит в разряд Final-релизов.

5.2.4.1 Порядок именования релизов

Название любого релиза состоит из четырёх цифр X.Y.W.Z, где:

- X – порядковый номер изменения платформы;
- Y – порядковый номер планового релиза с новой функциональностью;
- W – порядковый номер внепланового релиза с новой функциональностью, исправления продуктивной версии;
- Z – порядковый номер выданного Minor-релиза.

Если в ходе приемочного тестирования (до установки Major-релиза в продуктивную среду) были выявлены критичные дефекты, их исправление проводится в рамках Minor-релиза. Этот релиз получает метку X.Y.W, совпадающую с меткой Major-релиза, а индекс Z инкрементируется на 1.

Если ведется разработка продукта с адаптацией под различные типы платформ, в метку название релиза вносится дополнительный параметр Platform. Соответственно, метка название релиза меняется на X.Y.W.Z-Platform, где Platform – тип рабочей платформы продукта.

Если продукт состоит из отдельных модулей, подразумевающих возможность их отдельного использования или функционирования, политика их именования строго совпадает с общепринятой политикой именования продукта. Все используемые в продукте компоненты при выдаче очередной Major-версии продукта Заказчику имеют одинаковые метки X.Y.W, которые гарантированно определяют принадлежность данных версий компонентов версии Major-релиза продукта, выдаваемой Заказчику.

5.2.4.2 Система контроля версий

Для работы с исходными кодами и документацией используется система Confluence.

Такая система имеет древовидную структуру хранения данных проекта, которая позволяет:

- отслеживать все изменения;
- организовывать одновременную работу группы разработчиков над общим программным кодом продукта;
- организовывать одновременную работу над несколькими версиями продукта, восстановление измененных и удаленных частей исходных кодов продукта.

На Рис. 3 изображена схема контроля версий с использованием Confluence.

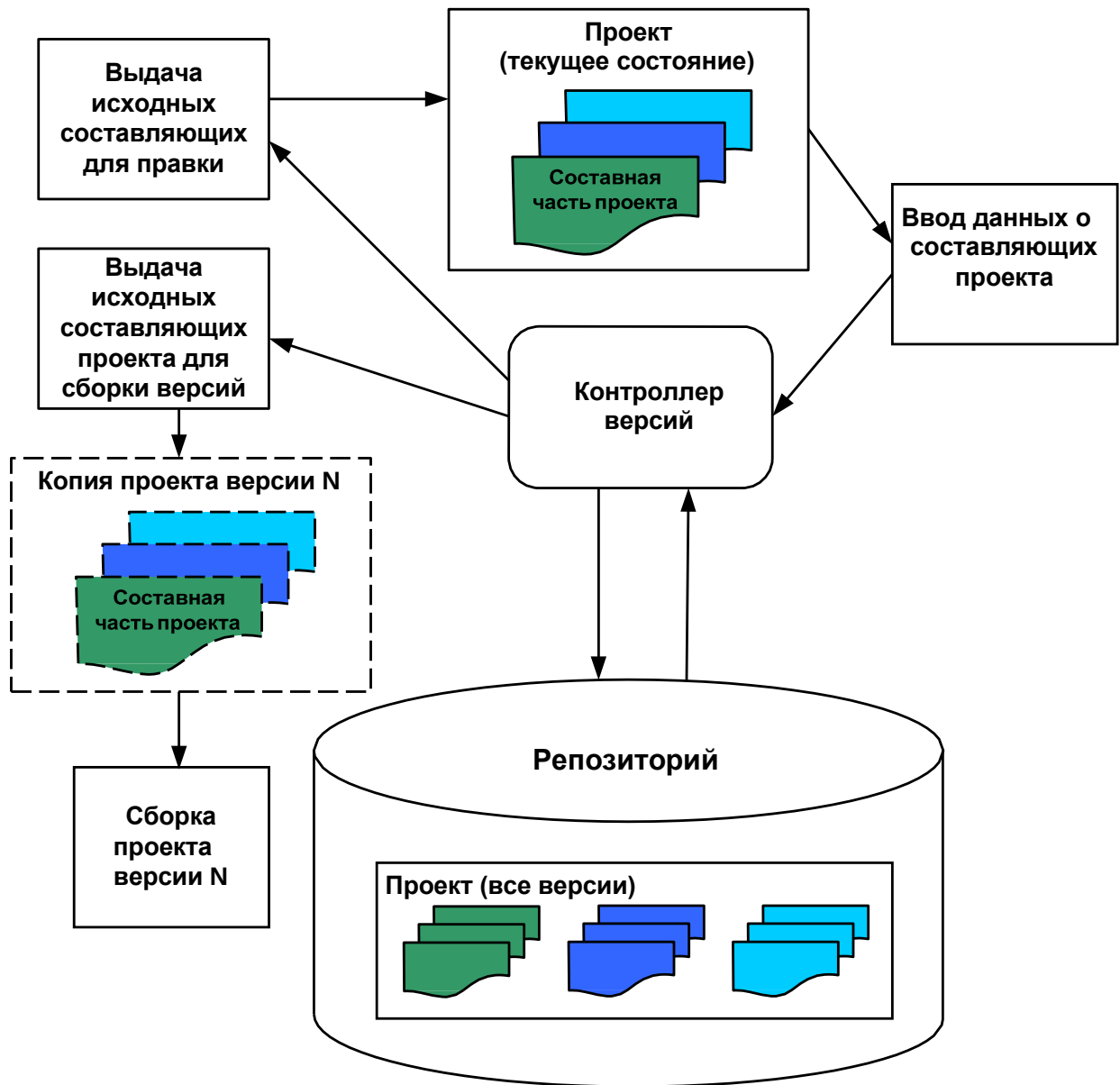


Рис. 3 – Схема контроля версий ПО

Для каждой информационной системы в Confluence создается отдельный, соответствующий этой системе проект.

При работе с исходными кодами участники проекта придерживаются следующих правил Confluence:

- Работа над текущей версией Major-релиза идет в стволе (HEAD) дерева проекта.
- В момент фиксации Candidate-релиза на его основе создается ветка проекта, в которой идет дальнейшая работа в рамках исправления критичных ошибок, выявленных в ходе тестирования.
- Название ветки однозначно идентифицирует разрабатываемую версию программного продукта и имеет вид: *<имя проекта/компонента> X.Y.W.*

- Для каждой версии Major-релиза создается ветка документации, содержащая весь перечень сопроводительных документов Системы.
- Все исправления для данной ветки дерева проекта ведутся в рамках создания Minor-версий релизов продукта. При этом отдельные ветки для Minor-версий не создаются.
- Когда ветка проекта создана, ведется работа над следующей Major-версией релиза продукта.
- Все исправления, сделанные в рамках разработки Minor-версий продукта в соответствующей ветке, должны быть проанализированы на предмет их переноса в другие ветки/ствол дерева проекта – Major-версии релизов программного продукта.

Соблюдение этих правил в работе с деревом проекта, а также особенности архитектуры системы контроля версий позволяют в любой момент времени получать актуальные версии различных релизов информационных систем.

Более подробно процессы тестирования программных продуктов представлены в разделе 5.2.5 «Тестирование ПО».

5.2.4.3 Резервирование релизов

Каждый выдаваемый Заказчику релиз, включая исходные коды и документацию для всех компонентов данного релиза, в обязательном порядке сохраняется в отдельном хранилище данных. Кроме того, при необходимости, создаются дополнительные резервные копии на оптических или иных носителях.

5.2.5 Тестирование ПО

Для тестирования ПО используется широкий набор программного инструментария, начиная со свободно распространяемых средств тестирования и заканчивая внутренними разработками Компании. Например, для решения задач по автоматизации процесса тестирования, который бы позволял быстро и точно адаптироваться к специфике различных программных решений, была разработана своя тестовая платформа.

В основе этой платформы лежит технология Node.JS, позволяющая избавиться от платформенных ограничений. Это, в свою очередь, дает возможность отделить тестовую среду от среды разработки без потери функциональности.

При реализации компонентов тестовой платформы были учтены жесткие требования к большим нагрузкам, поэтому платформа может использоваться и для Функционального, и для Нагрузочного тестирования.

Можно перечислить следующие основные преимущества разработанной в Компании тестовой платформы:

- типизация – написание шаблонов, упрощающих написание серии (автоматизированных) тестов;
- поддержка большинства используемых технологий посредством реализованных интерфейсов и адаптеров;
- возможность добавления различных модулей с поддержкой дополнительных сервисов;
- запуск тестов полностью в автоматическом режиме;
- поддержка возможности регрессивного тестирования с расширенным протоколированием результатов;
- поддержка нагрузочного тестирования;
- автоматическая генерация отчетов о тестировании.

Для обеспечения качества тестирования специалистами Компании разработаны комбинации тестов новой функциональности и регрессионных тестов. Кроме функционального тестирования периодически проводятся тесты производительности (performance), тестирование общей надежности (stability), тестирование поведения при перегрузке (stress-test), тесты пользовательского интерфейса (GUI).

5.2.6 Процесс управления дефектами

Для регистрации дефектов ПО и их исправления применяется формализованная методика.

В качестве основного инструментального средства управления дефектами используется открытое программное обеспечение Bugzilla.

Процесс управления дефектами изображен на Рис. 4.

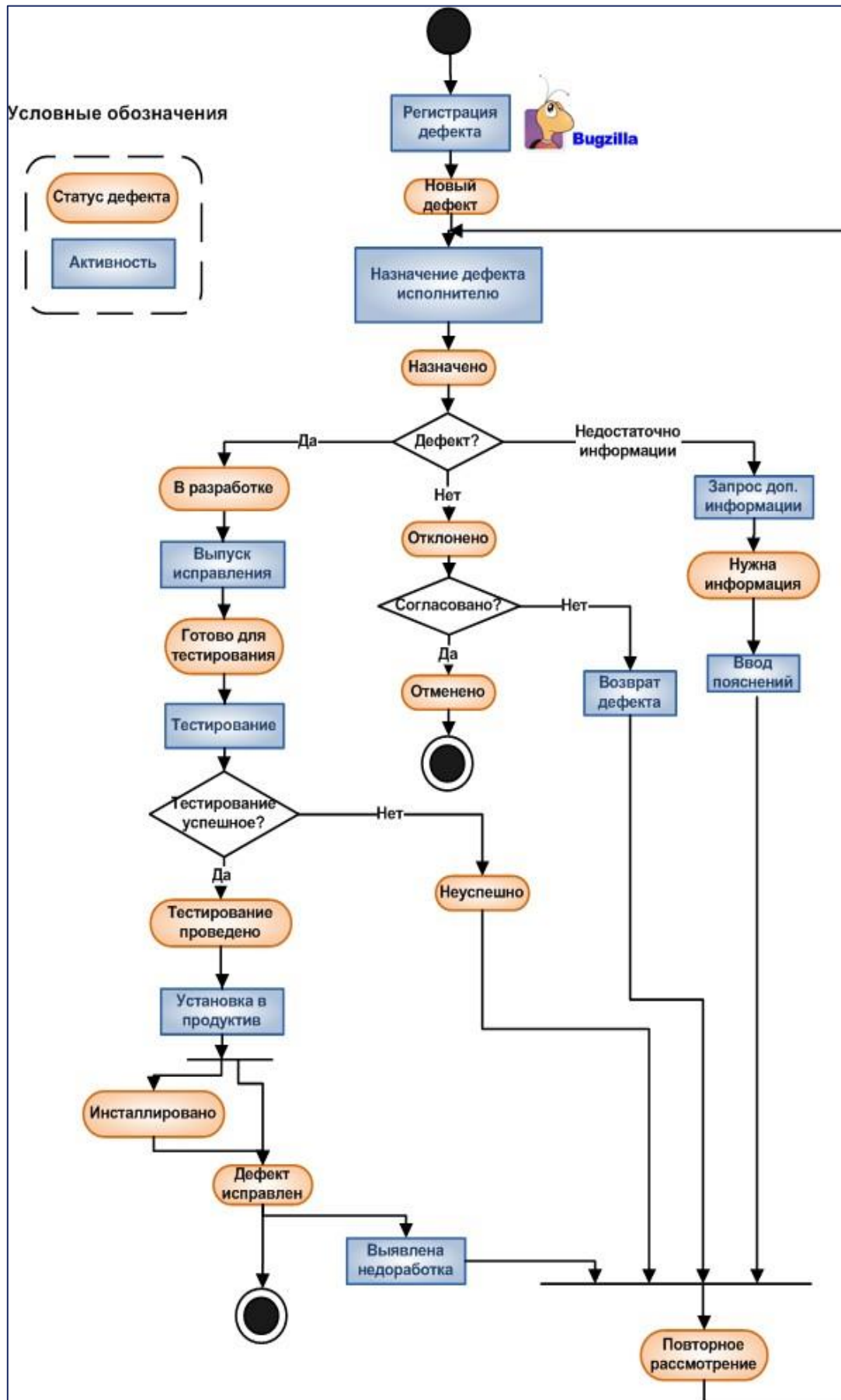


Рис. 4 – Процесс управления дефектами

5.3 Завершение проекта

Финальная стадия заключается в сдаче ПО Заказчику и перевод его на гарантийное обслуживание. Сдачу-приёмку готового продукта принимает комиссия, состав которой утверждает Заказчик. Члены комиссии проводят испытания в соответствии с положениями и проверками, включенными в утвержденную Программу и методику испытаний. Сдача ПО оформляется протоколом приемочных испытаний и актом Сдачи-приёмки.

Приемочные испытания могут проходить в два этапа. Необходимость предварительных и окончательных испытаний устанавливается по согласованию с Заказчиком.

На находящиеся в эксплуатации программные продукты распространяются гарантийные обязательства Исполнителя. Гарантийный срок сопровождения составляет 12 месяцев, начиная с даты приёмки работ. По соглашению с Заказчиком этот срок может изменяться. В течение гарантийного периода все выявленные дефекты Исполнитель устраняет бесплатно.

Постгарантийное обслуживание регламентируется отдельным договором. Такой договор предусматривает заключение двухстороннего соглашения об уровне сервиса.