

УТВЕРЖДЕН  
ru.red-soft.00001-01 37 01

ГЕМБАФЕЙС (GEMBAFACE)

Описание жизненного цикла и сопровождение продукта

ru.red-soft.00001-01 37 01

Листов 16

Инд. № подл.	Подп. и дата	Взам. инв. №	Инд. № дубл.	Подп. и дата

## **АННОТАЦИЯ**

ГЕМБАФЕЙС (GEMBAFACE) это программа для распознавания лиц. Предназначен для формирования баз данных поисковых образов (биометрических данных) по фотографиям людей и последующего поиска по предъявленной фотографии. Программа предполагает информационное взаимодействие со смежными информационными системами по технологии REST API.

В настоящем документе представлено описание системы управления жизненным циклом разработки и сопровождения ГЕМБАФЕЙС, которая используется в РЕД СОФТ при разработке ГЕМБАФЕЙС.

**СОДЕРЖАНИЕ**

1. ПО для обеспечения жизненного цикла и сопровождения . . . . .	3
1.1. Доступ к системе контроля версий . . . . .	3
1.2. Возможности системы контроля версий Git . . . . .	3
1.3. Возможности системы жизненного цикла GitLab . . . . .	4
1.4. Возможности системы автоматизации развертывания и управления приложениями Docker . . . . .	4
1.5. Возможности системы оркестрации Docker Compose . . . . .	5
2. Модель работы . . . . .	6
2.1. Жизненный цикл задач . . . . .	8
3. Система отслеживания ошибок . . . . .	9
4. Тестирование, отладка и оценка качества . . . . .	11
5. Внедрение и сопровождение . . . . .	12
6. Требования к персоналу . . . . .	13
Перечень терминов . . . . .	14

## **1. ПО ДЛЯ ОБЕСПЕЧЕНИЯ ЖИЗНЕННОГО ЦИКЛА И СОПРОВОЖДЕНИЯ**

Для обеспечения жизненного цикла разработки и сопровождения ГЕМБАФЕЙС используется широкий ряд различного ПО.

1.1. При разработке ГЕМБАФЕЙС для обеспечения контроля версий применяется программный продукт Git, который является свободно распространяемой по лицензии GNU/GPL v2 распределенной системой управления версиями. В РЕД СОФТ используется версия Git не ниже 2.0.

1.2. Для управления центральным репозитарием используется система GitLab, специальный инструмент жизненного цикла DevOps с открытым исходным кодом, представляющий систему управления репозиториями кода для Git. В текущий момент для разработки ГЕМБАФЕЙС используется версия GitLab Community Edition 13.1.3.

1.3. Для создания версий для тестирования и релизов используется Docker, это специализированное программное обеспечение для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации. Для разработки ГЕМБАФЕЙС используется версия Docker не ниже 17.

1.4. Для удобного управления несколькими контейнерами используется система оркестрации контейнеров Docker Compose. В основном используется самая актуальная версия системы.

1.5. Для фиксации и работы с задачами и ошибками системы используется система АСУП.

1.6. Для написания и изменения документации используется продукт Prelum.

### **1.1. Доступ к системе контроля версий**

В РЕД СОФТ доступ к репозиториям системы Git и к системе GitLab обеспечивается из контролируемых зон объектов РЕД СОФТ удаленно с использованием шифруемых протоколов Secure Shell (SSH), HTTPS. Доступ разработчиков обеспечивается внутри корпоративной виртуальной частной сети (VPN), организуемой сертифицированными средствами межсетевого экранирования и построения VPN каналов.

### **1.2. Возможности системы контроля версий Git**

Основными возможностями системы Git являются:

1) Возможность расширения функционала. Система спроектирована как набор программ, специально разработанных с учетом их использования в сценариях, это позволяет создавать специализированные системы контроля версий на базе Git и пользовательские интерфейсы;

2) Сохранение полной истории изменений файлов с возможностью возврата к предыдущим версиям. В истории сохраняются контрольные суммы изменений, поэтому она устойчива от

внесения правок, а сам репозиторий представляет собой не сколько файлы, сколько изменения файлов;

- 3) Система поддерживает очень быстрое создание веток, это в том числе влияет на модель разработки продукта при использовании системы Git;
- 4) Синхронизация изменений между пользователями и автоматическое слияние изменений;
- 5) Репозиторий Git является распределенным, а потому устойчивым от потерь;
- 6) Возможность фиксации изменений в ветках. Возможность пометки определенных изменений тегами.

### **1.3. Возможности системы жизненного цикла GitLab**

GitLab это интегрированный продукт, который автоматизирует весь жизненный цикл разработки программного обеспечения. GitLab расширяет возможности Git. Ключевые возможности GitLab:

- 1) Организация публичных и частных репозиторий;
- 2) Облачный хостинг;
- 3) Неограниченное число пользователей, расширенное управление правами, группами;
- 4) Импорт проектов;
- 5) Вики-страницы;
- 6) Интеграция по API;
- 7) Проставление лейблов и вет;
- 8) Использование поиска и шаблонов;
- 9) Комментирование, объединение;
- 10) Встроенные возможности для CI / CD автоматизации релизов - GitLab-CI;
- 11) Отслеживание изменений и прогресса;
- 12) Отслеживание времени;
- 13) Запросы на слияние кода;
- 14) Возможность проведения рецензирования кода.

### **1.4. Возможности системы автоматизации развертывания и управления приложениями Docker**

Для ускорения и одновременно упрощения и улучшения качества продуктов в РЕД СОФТ используется система контейнеризации Docker. Эта система используется для развертывания ГЕМБАФЕЙС в тестовой, разработки и в продуктовой средах.

Docker - это программная платформа для быстрой разработки, тестирования и развертывания приложений. Система контейнеризации Docker упаковывает ПО в стандартизованные блоки,

которые называются контейнерами. Каждый контейнер включает все необходимое для работы приложения: библиотеки, системные инструменты, код и среду исполнения. Благодаря Docker можно быстро развертывать и масштабировать приложения в любой среде.

Ключевые возможности Docker:

1) Минимальное потребление ресурсов, по сравнению с виртуальной машиной. Контейнеры не виртуализируют всю операционную систему (ОС), а используют ядро хоста и изолируют программу на уровне процесса. Последний потребляет намного меньше ресурсов локального компьютера, чем виртуальная машина.

2) Быстрая доставка контейнеров/образов. Быстрое развертывание ПО. Легкая переносимость - приложения на основе контейнеров Docker можно эффективно и быстро переносить в разные системы, даже разные операционные системы.

3) Изоляция ПО. Процессы эффективно изолируются друг от друга, также ОС изолируется от контейнеров, что минимизирует риски нарушения безопасности и работоспособности инфраструктуры.

4) Простота работы. Упрощает задачи настройки, доставки, развертывания и масштабирования ПО. Повышает качество установки ПО и минимизирует человеческие ошибки.

5) Стандартизация операций. Все операции по работе с контейнерами и образами Docker стандартизированы.

6) Поддерживается системами непрерывной интеграции. В частности поддерживается системой GitLab-CI.

### **1.5. Возможности системы оркестрации Docker Compose**

К основным возможностям системы оркестрации Docker Compose можно отнести:

1) Простота настройки и использования. Все настройки хранятся в отдельной папке с одним файлом конфигурации - `docker-compose.yml`. Управление сервисами: запуск, установка, настройка происходят простыми операциями и занимают минимальное время. Также возможно при этом организовывать сложные взаимосвязанные системы на базе нескольких приложений описанием в одном файле настроек.

2) Возможно создание нескольких изолированных сред на одном хосте, что также полезно при масштабировании систем.

## 2. МОДЕЛЬ РАБОТЫ

Общий процесс подготовки новых версий ГЕМБАФЕЙС в целом соответствует схеме (см. рис. 1):

2.1. Актуализация исходного кода (см. рис. 1 шаг 1) - осуществляется клонированием репозитория продукта или обновлением локального репозитория Git в основной ветке разработки (master). Этот этап осуществляется или консольным клиентом Git или любым другим клиентом Git.

2.2. Разработчик создает новую ветку от актуализированной основной ветки разработки для внесения изменений в исходный код продукта средствами клиента Git.

2.3. Разработчик вносит изменения в исходный код программы (см. рис. 1 шаг 2) любыми удобными для себя средствами разработки.

2.4. Разработчик фиксирует изменения и отправляет их в удаленный репозиторий Git на сервере GitLab (см. рис. 1 шаг 1), создавая при этом в удаленном репозитории новую ветку - копию локальной ветки исходного кода, созданной в пункте п. 2.2.

2.5. Разработчик локально тестирует все изменения в версии программы.

2.6. Когда накопится достаточно изменений для новой версии в нескольких ветках или в одной новой ветке, данные изменения сливаются в главную ветку master. Также в случае критических исправлений все изменения поступают в главную ветку проекта. Без тага подготавливаемая в ветке master версия становится основным кандидатом в релиз.

2.7. Система GitLab-CI для проекта ГЕМБАФЕЙС настроена таким образом, что при поступлении новых изменений в ветку master запускается автоматический пайплайн на сборку образа Docker новой версии ПО. Система ГЕМБАФЕЙС состоит из нескольких репозитариев нескольких различных компонент. Так что слитие (merge) изменений в главную ветку (master) любого из проектов ГЕМБАФЕЙС продуцирует автоматическое создание Docker образа с готовым ПО соотв. проекта (см. рис. 1 шаг 3).

2.8. В случае если произошла ошибка сборки в группу разработки проекта Telegram шлется уведомление об ошибке при сборке проекта.

2.9. Успешно созданные Docker образы компонент системы ГЕМБАФЕЙС автоматически регистрируются в репозиторий образов Docker (см. рис. 1 шаг 4). Новые Docker образы компонент системы ГЕМБАФЕЙС регистрируются в репозитории сборки Docker под тагами master (по имени ветки для которой строился проект).

2.10. В тестовой среде происходит настройка запуска готового проекта ГЕМБАФЕЙС (см. рис. 1 шаг 5). Для этого из репозитория образов Docker выкачиваются последние (актуальные) версии (версии помеченные тагом master) всех компонент системы ГЕМБАФЕЙС. Делается это командой docker pull в ручном режиме.

2.11. Тестовая среда ГЕМБАФЕЙС запускается посредством ранее настроенной системы оркестрации Docker Compose.

2.12. Тестовое приложение ГЕМБАФЕЙС, состоящее из нескольких компонент, и они же

кандидаты на новые версии тестируются (см. рис. 1 шаг 5). В случае выявленных проблем все шаги повторяются сначала для устранения ошибок и доработки функционала.

2.13. В случае если приложение готово к созданию версии, все ошибки устранены, все требования к новой версии выполнены: все репозитории ветки master всех компонент системы ГЕМБАФЕЙС фиксируются тагом с номером версии в формате «vM.m.B», где M - главный номер версии, m - младший номер версии, B - «build», используется для указания минимальных изменений в проекте. Версии компонент повышаются.

2.14. Система GitLab-CI для проекта ГЕМБАФЕЙС настроена таким образом, что при фиксации изменений в тагах запускается автоматический пайплайн на сборку образа Docker новой версии, но уже релиза с именем версии указанной в таге шаги 3-4 повторяются (см. рис. 1). В репозитории образов Docker фиксируются новые версии образов Docker отмеченные тагом в формате «vM.m.B». Это релизные (продуктовые) сборки компонент системы ГЕМБАФЕЙС и кандидаты на установку в продуктивную среду.

2.15. Данные образы поступают на установку в продуктивную среду выполнения системы ГЕМБАФЕЙС (см. рис. 1 шаг 6), в ручном режиме, аналогично тому как настраивается и запускается тестовая среда см. п. 2.11.

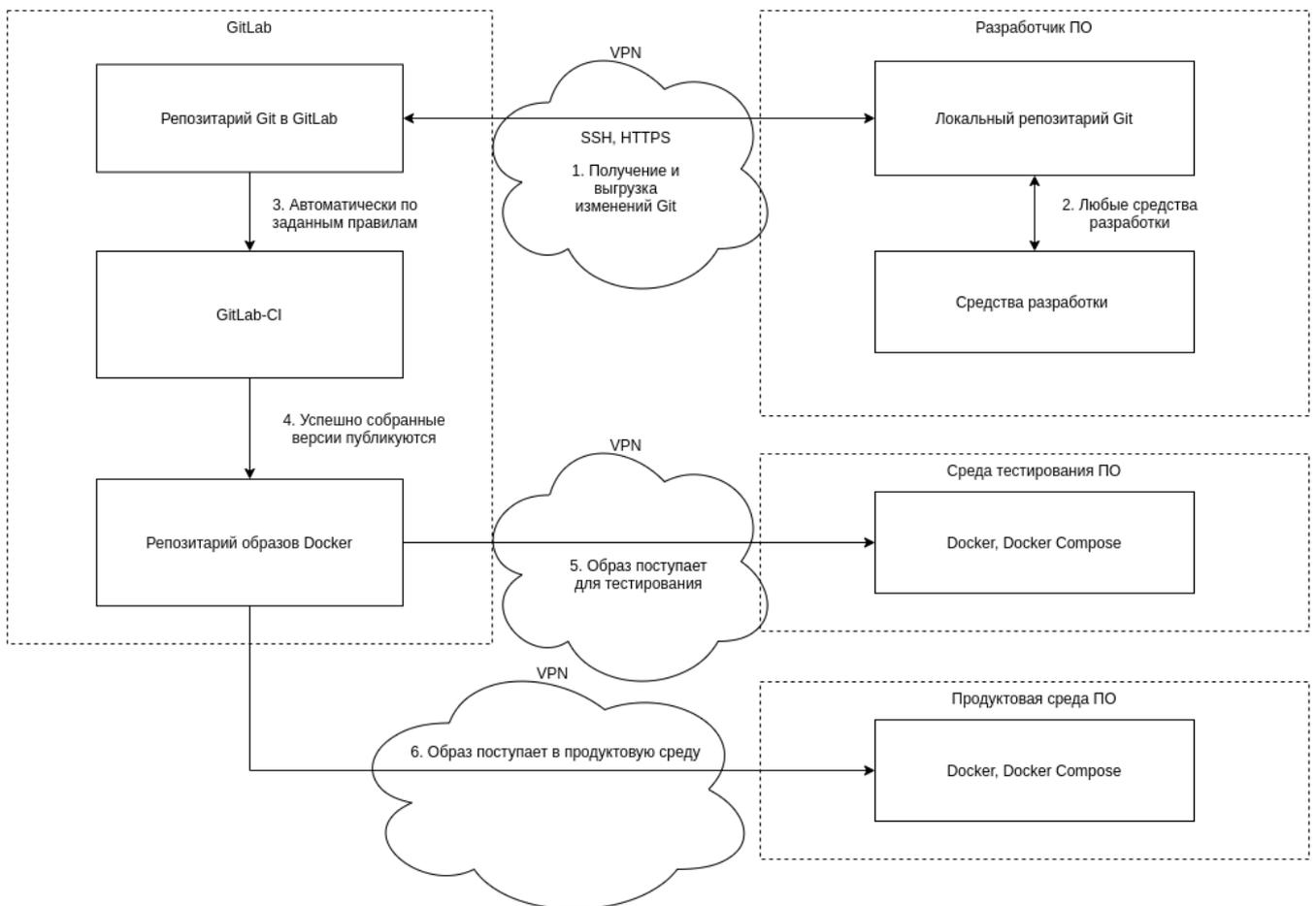


Рисунок 1

## **2.1. Жизненный цикл задач**

В применяемой при разработке ГЕМБАФЕЙС используется следующая схема жизненного цикла задач:

2.1.1. Регистрация ошибки или требований к доработке системы в задаче АСУП. Прием заявок на доработку, а также регистрация ошибок организована через АСУП.

2.1.2. Далее задачи распределяются по исполнителям, а также им назначается при необходимости срок выполнения.

2.1.3. В момент когда разработчик приступает к выполнению задачи, задача переводится со статуса «Открытый» в статус «В работе».

2.1.4. При завершении задачи задача переводится со статуса «В работе» на статус «На проверке». Тогда же задача проверяется тестировщиком или аналитиком или заказчиком на тестовом стенде.

2.1.5. В случае успешного выполнения задачи она переводится на статус «Закрыт». Если же при проверке задачи были выявлены ошибки, задача вновь переводится на статус «В работе».

### 3. СИСТЕМА ОТСЛЕЖИВАНИЯ ОШИБОК

При разработке ГЕМБАФЕЙС использована система отслеживания ошибок АСУП, предназначенная для учета и контроля ошибок, найденных в программах, пожеланий пользователей, а также для отслеживания процесса устранения этих ошибок и выполнения или невыполнения пожеланий.

Программа АСУП обеспечивает контроль исполнения любых производственных задач, заявок, документов организационно-распорядительного характера на стадии разработки и исполнения.

Программа обеспечивает в режиме реального времени отслеживание закрепленных за ответственными лицами задач, что делает ее незаменимой для руководителей при организации эффективного труда сотрудников, а также при организации взаимодействия сотрудников между собой.

Программа АСУП позволяет работать со справочниками данных. Справочники программы, используются для исключения заполнения полей вручную. Из справочника могут быть взяты следующие данные:

- статус заявки;
- автор записи;
- тип записи;
- название подсистемы;
- приоритет;
- дата обнаружения;
- тип обращения;
- срок исполнения заявки.

Использование справочников позволяет избежать ошибок при наборе реквизитов заявки, так как в справочниках хранится достоверная информация по заполнению полей в заявке. Если справочник был изменен, то при первом же запуске программы система сама автоматически обновит этот справочник.

Задача регистрируется в рамках работы линии технической поддержки пользователей. Это может быть проблема, возникшая у пользователя или сообщение об ошибке от системы, обнаруженное пользователем в процессе эксплуатации ГЕМБАФЕЙС. Так же задача может возникнуть в рамках работы группы разработки ПО или анализа ПО при его согласовании с заказчиком продукта.

Далее, созданная задача поступает на рассмотрение руководителей отделов и ведущих аналитиков. Исходя из контекста, они принимают решение об отказе или назначении в работу поступившей задачи.

Принятая в работу задача поступает в группу тестирования для подтверждения факта обнаруженной ошибки или непосредственно в группу аналитики или разработки, если такое подтверждение не требуется. Если ошибка не будет подтверждена в результате тестирования, она отклоняется.

Подтвержденная тестировщиками ошибка поступает для согласования и утверждения в группу аналитики и далее непосредственным разработчикам программного продукта.

После доработки или разработки вновь задача возвращается на тестирование, и если будет подтверждено исправление обнаруженной ошибки или корректная работа созданного компонента, задача закрывается.

#### **4. ТЕСТИРОВАНИЕ, ОТЛАДКА И ОЦЕНКА КАЧЕСТВА**

После каждой итерации разработки проводится тестирование новой функциональности системы ГЕМБАФЕЙС, и производится устранение ранее найденных ошибок.

Тестирование проводится лицами, ответственными за проведение тестирования программной продукции (отдел тестирования). Для тестирования и отладки программной продукции в Ред Софт обновляется стенд для тестирования. По результатам тестирования осуществляется устранение ошибок и осуществляется (при необходимости) доработка программного обеспечения.

При тестировании Системы на соответствие требованиям заказчика продукта и ее отладке выполняется:

- обновление тестового стенда;
- проводится тестирование алгоритмов ГЕМБАФЕЙС (автоматически, отчет о качестве фиксируется в артефактах Gitlab);
- проводится тестирование ГЕМБАФЕЙС тестировщиком;
- устраняются выявленные недостатки, обнаруженные в ходе тестирования;
- собирается конечная версия сборки и обновляется стенд;
- производится корректировка программной документации.

Сборка программного кода производится автоматически при регистрации изменений программного кода в ветке репозитория master и для коммитов отмеченных тагом с номером версии. Обновление тестового стенда так же производится автоматически при каждом коммите в ветку разработки - master (см. разд. 2).

Каждая сборка ГЕМБАФЕЙС имеет уникальный идентификатор - номер версии.

Перед передачей итоговой версии ГЕМБАФЕЙС потребителю проведение процесса тестирования является обязательным.

## **5. ВНЕДРЕНИЕ И СОПРОВОЖДЕНИЕ**

В период эксплуатации и сопровождения системы ГЕМБАФЕЙС Ред Софт оказывает помощь в установке, настройке, устранении недостатков в работе программного обеспечения, а также компания осуществляет обучение персонала, эксплуатирующего программное обеспечение.

Сопровождение Системы включает в себя такие работы как:

- исправление ошибок и устранение неполадок, возникающих при работе пользователей системы;

- оптимизация работы программы при различных условиях эксплуатации;
- обновление и доработка модулей Системы по требованиям заказчика продукта;
- настройка резервного копирования данных ГЕМБАФЕЙС;
- профилактические работы по обслуживанию Системы и ее компонентов;
- актуализация программной документации.

Техническая поддержка Системы осуществляется в формате консультирования пользователей и администраторов ГЕМБАФЕЙС по вопросам установки, переустановки, администрирования и эксплуатации по каналам связи (телефону, электронной почте) или письменно по запросу.

## 6. ТРЕБОВАНИЯ К ПЕРСОНАЛУ

Администратор системы должен владеть:

- общими принципами функционирования Системы, способами архивации и аварийного восстановления БД;
- принципы работы, установки и настройки используемых операционных систем;
- знать основы администрирования используемых СУБД (настройка учетных записей, настройка прав доступа);
- знать основы построения локальной вычислительной сети, уметь настраивать сетевое оборудование, обладать умениями и навыками установки и настройки конфигурации рабочего места пользователя;
- обладать навыками разрешения аппаратно-программных конфликтов в используемых операционных системах (настройки сети и сетевых протоколов);
- уметь управлять распределением прав пользователей Системы, уметь разрешать конфликты, связанные с настройкой рабочего места пользователя.

Специалисты группы технической поддержки должны знать:

- настройку программной и аппаратной части;
- обладать знаниями и умением классифицировать и устранять возникающие ошибки.

Администратор и специалисты группы технической поддержки должны ознакомиться с эксплуатационной документацией.

Все пользователи Системы должны быть «уверенными пользователями» операционной среды, базовыми знаниями работы с тестовыми редакторами, ознакомлены с руководством пользователя.

Обучение специалистов и пользователей может выполняться:

- самостоятельно с использованием разработанной документации;
- путем консультаций, согласно положениям по сопровождению и поддержке зарегистрированных пользователей.

**ПЕРЕЧЕНЬ ТЕРМИНОВ**

Термин	Определение
1. <b>API</b>	программный интерфейс приложения, интерфейс прикладного программирования.
2. <b>CI / CD</b>	практика разработки программного обеспечения, которая заключается в постоянном слиянии рабочих копий в общую основную ветвь разработки (до нескольких раз в день) и выполнении частых автоматизированных сборок проекта для скорейшего выявления потенциальных дефектов и решения интеграционных проблем.
3. <b>DevOps</b>	методология активного взаимодействия специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимная интеграция их рабочих процессов друг в друга для обеспечения качества продукта.
4. <b>Docker</b>	программное обеспечение для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесен на любую Linux-систему с поддержкой cgroups в ядре, а также предоставляет среду по управлению контейнерами.
5. <b>GEMBAFACE</b>	программа (набор сервисов) для распознавания лица, разработки Ред Софт.
6. <b>Git</b>	распределённая система управления версиями.
7. <b>GitLab</b>	веб-инструмент жизненного цикла DevOps с открытым исходным кодом, представляющий систему управления репозиториями кода для Git с собственной вики, системой отслеживания ошибок, CI/CD пайплайном и другими функциями.
8. <b>HTTPS</b>	расширение протокола HTTP для поддержки шифрования в целях повышения безопасности. Данные в протоколе HTTPS передаются поверх криптографических протоколов TLS или устаревшего в 2015 году SSL.
9. <b>Prelum</b>	программа предназначенная для написания документации по ГОСТ.

Термин	Определение
10. <b>Secure Shell (SSH)</b>	сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов).
11. <b>Telegram</b>	кроссплатформенный мессенджер, позволяющий обмениваться сообщениями и медиафайлами многих форматов.
12. <b>VPN</b>	обобщённое название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети (например Интернет).
13. <b>gamba-face</b>	альтернативное название программы GEMBAFACE.
14. <b>АСУП</b>	система основанная на Redmine для управления проектами и задачами в компании Ред Софт.
15. <b>ГЕМБАФЕЙС</b>	русскоязычное название программы GEMBAFACE.
16. <b>Канбан</b>	система организации производства и снабжения, позволяющая реализовать принцип «точно в срок».
17. <b>Канбан-доска</b>	является одним из инструментов, который может использоваться при внедрении метода управления разработкой «канбан».
18. <b>Контейнеризация</b>	(иначе, виртуализация на уровне операционной системы, контейнерная виртуализация, зонная виртуализация) метод виртуализации, при котором ядро операционной системы поддерживает несколько изолированных экземпляров пространства пользователя вместо одного. Эти экземпляры (обычно называемые контейнерами или зонами) с точки зрения пользователя полностью идентичны отдельному экземпляру операционной системы.
19. <b>ОС</b>	комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.
20. <b>Облачный хостинг</b>	тип веб-хостинга, который использует несколько разных серверов для того, чтобы распределять нагрузку и максимально увеличивать время безотказной работы.
21. <b>ПО</b>	программа или множество программ, используемых для управления компьютером (ISO/IEC 26514:2008).

Термин	Определение
22. <b>Продуктовая среда</b>	конкретный экземпляр конфигурации аппаратного и программного обеспечения, предназначенный для работы в контролируемой среде. В данной среде продукт доступен пользователям.
23. <b>РЕД СОФТ</b>	отечественный поставщик решений и услуг в области информационных технологий.
24. <b>Рецензирование кода</b>	(иначе обзор кода, ревизия кода (англ. code review) или инспекция кода (англ. code inspection) — систематическая проверка исходного кода программы с целью обнаружения и исправления ошибок, которые остались незамеченными в начальной фазе разработки. Целью обзора является улучшение качества программного продукта и совершенствование навыков разработчика.
25. <b>Среда разработки</b>	интегрированная система, включающая в себя аппаратные средства, ПО, программно аппаратные средства, процедуры и документы, необходимые для разработки ПО.
26. <b>Тестовая среда</b>	конкретный экземпляр конфигурации аппаратного и программного обеспечения, предназначенный для тестирования работы в контролируемой среде.