**Жизненный цикл ПО**

**«Система развития кадрового потенциала РЕКОРД (импортонезависимая):**

**Обучение»**

Содержание

[1 История изменения документа 3](#_Toc57193265)

[2 Термины и сокращения 4](#_Toc57193266)

[3 Назначение системы 5](#_Toc57193267)

[4 Проектирование и конструирование 5](#_Toc57193268)

[5 Сборка …………………………………………………………………………………………….6](#_Toc57193269)

[6 Тестирование 7](#_Toc57193270)

[7 Менеджмент конфигурации системы 7](#_Toc57193271)

[8 Процесс решения проблем системы 7](#_Toc57193272)

[9 Информация о персонале 8](#_Toc57193273)

# История изменения документа

Таблица 1 – История изменения документа

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Дата | Автор | Первая редакция/Описание изменений |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Термины и сокращения

|  |  |
| --- | --- |
| Термин/сокращение | Определение |
| ИС | Информационная система |
| ИТ | Информационные технологии |
| КПЭ | Ключевой показатель эффективности |
| ОС | Операционная система |
| ПО | Программное обеспечение |
| ПО Системы | Разрабатываемое в рамках Проекта программное обеспечение, реализующее основной функционал Системы, в том числе функции защиты информации, администрирования, миграции и интеграции. |
| Система | Система развития кадрового потенциала РЕКОРД (импортонезависимая): Обучение |
| СУБД | Система управления базами данных |
| ИПР | Индивидуальный план развития |

# Назначение системы

Система представляет собой комплекс программно-технических средств, реализованных с использованием микросервисной архитектуры и свободно распространяемого ПО. Система предназначена предназначена для автоматизации бизнес-процессов обучения сотрудников на предприятиях.

Отдельные компоненты Системы могут взаимодействовать между собой в рамках сквозных процессов. Система, предоставляет пользователям возможности по выполнению ключевых функций:

В функциональные возможности входят:

* + актуализация требований и ресурсов обучения;
  + планирование обучения;
  + организация и проведение обучения;
  + дистанционное обучение;
  + отчетность и аналитика по обучениям;
  + средства администрирования;
  + управление ролями пользователей.

# Проектирование и конструирование

Для Системы выбрана эволюционная стратегия конструирования. В процессе развития Системы используется спиральная модель жизненного цикла и релизный подход при обновлении.

Обобщенные фазы спиральной модели Система выглядят следующим образом:

* + фаза сбора потребностей: в рамках фазы идет взаимодействие с Заказчиком, выявление потребностей в изменении Системы. Написание проектной документации.
  + фаза планирования релиза: в рамках фазы устанавливаются изменения, которые будут внесены в Систему в следующей версии Системы и установлены в продуктивный контур в рамках релиза. План релиза согласовывается с владельцем Системы.
  + фаза разработки: в рамках фазы проходит разработка заявленных доработок по проектной документации. Разработка ведется в контуре разработки. В результате разработки формируется сборка с внесенными изменениями и устанавливается на контур тестирования.
  + фаза тестирования: в рамках фазы осуществляется тестирование Системы на различных контурах:
* контур разработки
* контур тестирования
* контур регрессионного тестирования
* контура нагрузочного тестирования
* контур продуктива.
  + результаты тестирования разных контуров:
* тестирование в контуре разработки – решение о возможности установки сборки на тестовый контур;
* тестирование в контуре тестирования – решение о возможности включения изменения в сборку для регрессионного тестирования и контура нагрузочного тестирования;
* тестирование в контуре регрессионного и нагрузочного тестирования – решение о возможности установки изменений в продуктивный контур;
* тестирование продуктивного контура – решение о возможности выпуска релиза и сохранении изменений на продуктивном контуре.
* фаза приемо-сдаточных испытаний: в рамках фазы проводится демонстрация реализованных изменений, сбор замечаний и принятие решение о готовности перевода доработки на продуктивный контур.

# Сборка

Сборка приложения осуществляется из исходного кода. Исходный код хранится в системе управления версиями. Для формирования сборки приложения Системы используются инструмент GitLab Runer, для которого разработаны специализированные автоматизированные процедуры.

В результате работы инструмента GitLab Runer на выходе получаются:

* + готовые к установке приложения/компоненты Системы;
  + набор скриптов для модификации модели данных Системы.

# Тестирование

Для Системы применяются следующий виды тестирования:

Тестирование частной доработки – испытания, направленные на тестирование частных доработок системы, без проверки полного функционала Системы. Если частная доработка подразумевает под собой использование внешних систем, для тестирования применяются специализированные заглушки, имитирующие внешние системы. Интеграционное тестирование – испытания, направленные на выявление проблем взаимодействия отдельных компонентов Системы или взаимодействия с внешними системами.

Регрессионное тестирование испытания, направленные на подтверждение корректности поведения Системы в целом, подразумевающее сквозную проверку взаимодействия всех компонент Системы и взаимодействия с внешними системами. Нагрузочное тестирование – испытание информационной Системы в условиях прогнозируемой нормальной нагрузки и стрессовой нагрузки. Под величиной нагрузки понимается количество пользовательских запросов к Системе, которое она должна успевать обрабатывать, не превышая определенное исходными требованиями время отклика.

# Менеджмент конфигурации системы

Конфигурирование Системы осуществляется согласно внутриотраслевым стандартам управления изменениями информационных систем и хранением исходных кодов в системе контроля версий.

Процесс управления изменениями информационной Системы в общем виде содержит следующие шаги:

* + получение потребности в изменении Системы;согласование изменение Системы;
  + проектирование изменений Системы; разработка изменений Системы;
  + проверка качества изменения Системы; внедрение изменений Системы.

# Процесс решения проблем системы

Для решения проблем организована служба технической поддержки Системы. Проблема, связанная с Системой, является обращением. Типы обращений приведены в таблице ниже.

Таблица 1 – «Типы обращений»

|  |  |  |
| --- | --- | --- |
| № | Тип | Формат текста описания заявки |
| 11 | Инцидент (ошибка ПО, проблема смежных сервисов) | Несоответствие работы Системы утвержденным техническим решениям и эксплуатационной документации, появление окна с сообщением об ошибке |
| 22 | Запрос на изменение | Замечание/предложение по работе Системы (запрос на изменение/расширение функциональности) |
| 34 | Консультация | Просьба пользователя о пояснении работы в Системы, уточнение каких-либо непонятных моментов |
| 44 | Запрос на обслуживание | Запрос на администрирование справочников, изменение данных, удаление/корректировку документов |
| 55 | Заявка на предоставление доступа | Запрос на подключение/отключение пользователя, изменение прав доступа |
| 66 | Запрос на оказание разовых услуг | Запрос на выгрузку отчетов либо выполнение работ, не предусмотренных штатным функционалом Системы |

# Информация о персонале

Для эксплуатации системы предполагается наличие нескольких типов (ролей).

Администраторов Системы:

* + Системный администратор (не менее 3 человек);
  + Администратор БД (PostgreSQL) (не менее 2 человек);
  + Администратор веб – сервера (не менее 3 человек);
  + Администратор приложения (не менее 3 человек);

Один и тот же специалист может совмещать несколько типов(ролей) администраторов.

Системные Администраторы должны обладать следующими навыками:

* + администрирования ОС Linux – в рамках сертифицированных курсов;
  + администрирования платформы виртуализации VMware в рамках сертифицированных курсов;
  + администрирования системы резервного копирования в рамках сертифицированных курсов.
  + работы с инструментами контейнеризации и оркестрации (Kubernetes,Doker)

Администраторы БД (PostgreSQL) должны обладать следующими навыками:

* + администрирования PostgreSQL – в рамках сертифицированных курсов;
  + диагностики производительности СУБД – в рамках сертифицированных курсов.

Администраторы cервера приложений должны обладать следующими навыками:

* + администрирования веб - серверов Apache\Nginx – в рамках сертифицированных курсов.
  + работы с инструментами мониторинга (Grafana, Prometheus, Influx, Sentry)
  + работы с инструментами поиска,кэширования и хранилиша S3 (Elastic Search , Minio , Redis)

Администраторы приложения должны обладать следующими навыками:

* + работы в Системе на уровне пользователя
  + работы в Системе на уровне администратора
  + знания функциональных и технических возможностей Системы
  + анализа и решения инцидентов

Для развития системы требуются предполагается наличие нескольких типов (ролей):

* + Аналитик (не менее 4 человек);
  + Разработчик (не менее 10 человек);
  + Специалист по качеству (не менее 10 человек).

Аналитики должны обладать следующими навыками:

* + сбора и выявление требований;
  + разработки в BPMN нотации;
  + понимания процессов разработки ПО;
  + знания процессов управления развитием персонала.

Разработчики должны обладать следующими навыками:

* + разработки корпоративных приложений и систем (web) с использованием технологий JavaScript, TypeScript, NodeJS, ReactJS, MobX, Molecular;
  + оптимизации производительности и надежности систем;
  + работы с СУБД, способность по плану выполнения запроса предложить способы оптимизации;
  + работы с инструментами автоматической сборки проектов (Maven, Ant);
  + использования технологий и инструментов интеграции систем (Web services, SOAP,REST, JMS и т.п.).
  + работы с инструментами GitLab, Jira, Confluence

Специалисты по качеству должны обладать следующими навыками:

* + ручного и автоматического тестирования ИС, мобильных решений;
  + разработки сценариев тестирования, формирования отчетности;
  + знание принципов работы систем с web-интерфейсом;
  + работы с инструментами Jenkins, Cucmber, Selenium