Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ   
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

|  |
| --- |
| **Институт информационных технологий и анализа данных** |
| наименование института |
|  |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Допускаю к защите  Руководитель ООП |
|  |  | Д. Я. Постельник |
| подпись |  | И.О. Фамилия |

|  |
| --- |
| Применение систем ALM в Управлении информационными |
| технологиями Иркутского авиационного завода |
| наименование темы |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной квалификационной работе бакалавра

Программа бакалавриата:

Информационные системы и технологии

на предприятии / в машиностроении

по направлению подготовки

09.03.02 «Информационные системы и технологии»

|  |  |
| --- | --- |
| 0.041.00.00 ПЗ |  |
| обозначение документа |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Разработал студент группы ИСМб-16-1 |  |  |  | С.А. Базыль |
|  |  | подпись |  | И.О. Фамилия |
| Руководитель |  |  |  | Е.А. Барахтенко |
|  |  | подпись |  | И.О. Фамилия |
| Консультанты |  |  |  | Т.И. Дроздова |
|  |  | подпись |  | И.О. Фамилия |
|  |  |  |  |  |
|  |  | подпись |  | И.О. Фамилия |
| Нормоконтроль |  |  |  | Е.А. Барахтенко |
|  |  | подпись |  | И.О. Фамилия |

Иркутск 2020 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ   
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Институт информационных технологий и анализа данных** | | | |
| наименование института | |
|  | |
|  | |
| УТВЕРЖДАЮ | |
| Директор института | |
|  | Д.Я. Постельник |
| «27» 1 апреля 1 2020 г. | |

ЗАДАНИЕ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| на выпускную квалификационную работу студенту | | | | | Базыль Сергею | | | |
| Александровичу | | | | | | | | |
| группы | | ИСМб-16-1 |  | | | | | |
| 1. Тема работы: Применение систем ALM в Управлении информационными | | | | | | | | |
| технологиями Иркутского авиационного завода | | | | | | | | |
|  | | | | | | | | |
| Утверждена приказом по университету от | | | | 27.04.2020 | |  | № 734 |  |
| 1. Срок представления студентом законченной работы в ГЭК | | | | | | | 25.06.2020 г. | |
| 1. Исходные данные | | | | | | | | |
|  | 3.1 СТО 005-2015 «Система менеджмента качества. Учебно- | | | | | | | |
|  | методическая деятельность. Оформление курсовых проектов (работ) и | | | | | | | |
|  | выпускных квалификационных работ технических специальностей» | | | | | | | |
|  | 3.2 Материалы преддипломной практики | | | | | | | |
| 1. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов): | | | | | | | | |
|  | 4.1 | | | | | | | |
|  | 4.2 | | | | | | | |
|  | 4.3 | | | | | | | |
|  | 4.4 | | | | | | | |
|  | 4.5 | | | | | | | |
| 1. Перечень графического материала (с указанием обязательных чертежей) | | | | | | | | |
|  | 5.1 Презентационные материалы | | | | | | | |
| 1. Дополнительные задания и указания | | | | | | | | |
|  | 6.1 | | | | | | | |
|  | 6.2 | | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 7. Консультанты по работе с указанием вопросов, подлежащих решению | | | | | | |
|  | По безопасности жизнедеятельности | | | | |  |
|  |  | | | | |  |
|  |  | «\_16\_» \_\_июня\_\_\_\_\_2020 г. |  |  | Т.И. Дроздова |  |
|  |  |  | подпись |  | И.О. Фамилия |  |
|  |  | | | | |  |
|  |  | | | | |  |
|  |  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2020 г. |  |  |  |  |
|  |  |  | подпись |  | И.О. Фамилия |  |

Календарный план

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Разделы | Месяцы и недели | | | | | | | | | | | |
| апрель | | | | май | | | | июнь | | | |
| Введение |  |  | + | + | + |  |  |  |  |  |  |  |
| 1. |  |  |  |  |  | + |  |  |  |  |  |  |
| 2. |  |  |  |  |  |  | + |  |  |  |  |  |
| 3. |  |  |  |  |  |  |  | + |  |  |  |  |
| Безопасность жизнедеятельности |  |  |  |  |  |  |  |  | + |  |  |  |
| Заключение |  |  |  |  |  |  |  |  | + |  |  |  |
| Оформление пояснительной записки |  |  |  |  |  |  |  |  | + | + | + | + |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Дата выдачи задания | | | « | 27 | | » | | | апреля 2020 г. | | | | |  | |
|  | | | | | | | | | |  | | | | | |
|  | Руководитель работы | | | | | | | | |  | | | | Е.А. Барахтенко | |
|  |  | | | | | | | | | подпись | | | | И.О. Фамилия | |
|  |  | | | | | | | | |  | | | | | |
|  | Руководитель ООП | | | | | | | | |  | | | | Д.Я. Постельник | |
|  |  | | | | | | | | | подпись | | | | И.О. Фамилия | |
|  |  | | | | | | | | |  | | | | | |
|  | Задание принял к исполнению студент | | | | | | | | | |  | | | С.А. Базыль | |
|  |  | | | | | | | | | | подпись | | | И.О. Фамилия | |
|  | План выполнен |  | | | | | | | | | | | | | |
|  |  | (полностью, не полностью) | | | | | | | | | | | | | |
|  | Руководитель работы | | | « |  | | » |  | | | | 2020 г. |  |  | Е.А. Барахтенко |
|  |  | | |  |  | |  | дата | | | |  | подпись |  | И.О. Фамилия |

Содержание

[Введение 6](#_Toc43820173)

[1 Управление жизненным циклом приложений (ALM) 8](#_Toc43820174)

[1.1 Определение ALM 8](#_Toc43820175)

[1.2 Назначение системы 9](#_Toc43820176)

[2 Особенности системы ALM 10](#_Toc43820177)

[2.1 Основные аспекты систем ALM 10](#_Toc43820178)

[2.2 Роли участников в системе 11](#_Toc43820179)

[2.3 Отличие систем ALM от систем SDLC 13](#_Toc43820180)

[2.4 Преимущества в применении систем ALM 14](#_Toc43820181)

[2.5 Недостатки в применении систем ALM 15](#_Toc43820182)

[3 Программные решения в области ALM 17](#_Toc43820183)

[3.1 Microsoft Visual Studio Team System 17](#_Toc43820184)

[3.2 TeamForge 19](#_Toc43820185)

[3.3 Devprom ALM 20](#_Toc43820186)

[3.4 Kovair ALM Studio 22](#_Toc43820187)

[3.5 SwiftALM 23](#_Toc43820188)

[3.6 Open ALM 24](#_Toc43820189)

[3.7 SAP Solution Manager 25](#_Toc43820190)

[4 Применение системы ALM для управления разработкой приложения 27](#_Toc43820191)

[4.1 Выбор системы ALM 27](#_Toc43820192)

[4.2 Исходные данные 28](#_Toc43820193)

[4.3 Создание проекта и описание доступных функций платформы Devprom ALM 30](#_Toc43820194)

[5 Безопасность жизнедеятельности 44](#_Toc43820195)

[5.1 Характеристика объекта 44](#_Toc43820196)

[5.2 Анализ вредных и опасных факторов 45](#_Toc43820197)

[5.3 Приведение в норму производственного микроклимата 47](#_Toc43820198)

[5.3.1 Расчет комфортности производственной среды 49](#_Toc43820199)

[5.4 Электробезопасность 51](#_Toc43820200)

[5.5 Пожарная безопасность 52](#_Toc43820201)

[5.5.1 Мероприятия по эвакуации 53](#_Toc43820202)

[5.6 Выводы по разделу 54](#_Toc43820203)

[Заключение 55](#_Toc43820204)

[Список использованных источников 56](#_Toc43820205)

# Введение

Иркутский авиационный завод (ИАЗ) - это крупный промышленный комплекс Иркутской области, который входит в число самых современных и развивающихся предприятий по производству авиационной техники.

На территории ИАЗ находится большое количество различных отделов, каждое из которых выполняет свою, строго определенную задачу. Есть на заводе и отделы, которые занимаются разработкой программного обеспечения различного назначения и для определенных целей.

Но с каждым годом разрабатывать новое программное обеспечение становиться все сложнее. Поэтому всю разработку ПО стараются делить на различный уровень процессов (или дисциплин). К таким процессам можно отнести: анализ требований, проектирования программного обеспечения, программирование, тестирование ПО, внедрение и сопровождение.

Данные этапы жизненного цикла для каждого программного продукта поддерживаются различными системами, используется разный инструментарий для конкретных целей и задач, в каждом этапе принимают участие специалисты разных направлений. Все это усложняет разработку качественного и оптимизированного программного обеспечения.

Для облегчения разработки необходимо контролировать и координировать действия всех этапов разработки программного обеспечения. К сожалению, зачастую очень сложно объединить весь инструментарий по разработке программного обеспечения и заставить его обмениваться информацией между собой, сохранив при этом согласованность данных и семантику.

Именно здесь на помощь приходит система управления жизненным циклом приложений (ALM). Система ALM предоставит инструменты, процессы и технологии, которые будут направлены на повышение предсказуемости, согласованности и измеримости процесса разработки, а также поспособствует координации, управлению и синхронизации различных видов деятельности при разработке программного обеспечения.

Соответственно, интегрируя задачи разработки, совместной работы, коммуникации и управления знаниями и централизуя управление пользователями, проектами и процессами, ALM может хорошим решением в разработке программного обеспечения. Ведь ALM представляет собой комплекс для контроля и управления процессами на всех этапах разработки программного обеспечения и за его пределами, что является важным шагом к полной автоматизации и эффективности процесса разработки ПО.

В данной работе будет рассмотрено понятие системы ALM, какое назначение у данных систем, описаны основные особенности, достоинства и недостатки в применении. Исследуя данные системы, также будут рассмотрены наиболее популярные на данный момент имеющиеся на рынке ALM системы от различных разработчиков.

В практической части работы будет показано, как можно применить платформу ALM для организации работ по разработке программного обеспечения в отделах Управления информационными технологиями Иркутского авиационного завода.

# Управление жизненным циклом приложений (ALM)

## Определение ALM

ALM — это управление жизненным циклом приложений, которое включает в себя управление, разработку и обслуживание [1]. Помимо сказанного, в систему можно включить также следующие дисциплины: управление проектами, управление выпусками, архитектура программного обеспечения, тестирование, разработка требований и др.

В одной из научных статей приводится следующее определение системы ALM: Управление жизненным циклом приложения - это интегрированная система людей, процессов и инструментов, которые управляют ЖЦ приложения от концепции до выхода из эксплуатации [2].

Также нередко **управление жизненным циклом приложений (ALM) рассматривается как процесс выпуска программных средств с постоянно повторяющимся циклом следующих взаимосвязанных этапов** [3]**:**

* определение требований (Requirements);
* проектирование (Design);
* анализ проекта (Analysis);
* разработка (Development);
* тестирование (Testing);
* развертывание (Deployment);
* сопровождение (Maintenance).

Само понятие ALM как средство для создания программного обеспечения возникло более сорока лет назад, когда формировались основные методы по разработке программного обеспечения в промышленных масштабах. Однако больше всего усилий по решению задачи об автоматизации разработки программного обеспечения были направлены на создание такого набора инструментов, который был связан с программированием, т.к. он считался самым трудоемким процессом в разработке. И только к 80-ым годам 20 века, в связи с повышением сложности разработки проектов новых программ, ситуация с системами ALM стала значительно изменяться [3].

## Назначение системы

Основная идея, лежащая в основе систем ALM - создание общей инфраструктуры для разработки программного обеспечения. Такая инфраструктура поможет централизовать, организовывать и согласовывать все процессы и инструменты, которые связанны с разработкой.

Ведь разработка приложений в наши дни играет ключевую роль в развитии информационных технологий. Но разработка, как известно, непростой процесс. Она организована на взаимодействии и слаженной работе большого количества людей и применении различного инструментария, что в результате влияет на конечное качество продукта.

Ведь часто отсутствие координации и согласованности действий в разработке программного обеспечения и большое разнообразие инструментария для разработки приводит к тому, что на финальном этапе возникает большое количество ошибок перед выпуском приложения, также это приводит к серьезным финансовым затратам и создает большие временные издержки.

Правильно же организованная ALM-система позволяет [4]:

* сократить сроки вывода продуктов на рынок (разработчикам приходится заботиться лишь о соответствии их программ сформулированным требованиям);
* повысить качество, гарантируя при этом, что приложение будет соответствовать потребностям и ожиданиям пользователей;
* повысить производительность труда (разработчики получают возможность делиться передовым опытом разработки и внедрения);
* ускорить разработку благодаря интеграции инструментов;
* уменьшить затраты на сопровождение, постоянно поддерживая соответствие между приложением и его проектной документацией;

# Особенности системы ALM

## Основные аспекты систем ALM

Эксперт в области ИТ Дэвид Чеппел предостерегает от упрощенного взгляда на ALM, которое часто отождествляют лишь с жизненным циклом разработки программного обеспечения (Software Development Lifecycle, SDLC). Но ALM охватывает более широкий круг задач, рассматривая все аспекты существования такого корпоративного ресурса, как приложения [5].

Также, в его работе под названием "What is Application Lifecycle Management", рассказывается о трех основных областях, которыми характеризуется любая система ALM: руководство (Governance), разработка (Development) и эксплуатация (Operations) [6].

Все эти три аспекта показаны в виде горизонтальной линии жизненного цикла любого проекта на рисунке 2.1. Далее стоит описать каждый аспект поподробнее.

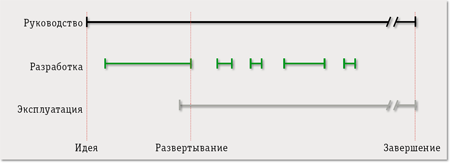


Рисунок 2.1 - Аспекты ALM

Руководство.

Целью руководства в ALM является обеспечение того, чтобы программный продукт удовлетворял потребностям бизнеса, в котором он создается. Руководство - это единственный аспект ALM, который существует на всем протяжении жизненного цикла. Его можно отнести к самому важному аспекту ALM. Если руководство будет осуществляться неверно, то и невозможно будет максимизировать ценности приложения для бизнеса.

Разработка.

Разработка, безусловно, является фундаментальной частью жизненного цикла каждого пользовательского приложения. Она происходит от момента появления идеи о создании нового программного продукта и заканчивается этапом выхода в эксплуатацию готового программного обеспечения.

Процессы, присущие разработке, могут применяться и после того, как готовое решение выйдет на рынок. Это необходимо для того, чтобы внести изменения в приложение или подготовить новую версию. В разработку необходимо включать следующие компоненты: определение требований, проектирование, кодирование и тестирование. Данные процессы реализуются в разработке в виде итераций, и такие итерации могут повторяться в проекте.

Эксплуатация.

Каждое приложение должно после ввод в эксплуатацию отслеживаться и поддерживаться разработчиками. Это является ключевым моментом в жизненном цикле программного обеспечения, ведь зачастую разрозненность между командой разработчиков и командой поддержки программного обеспечения создает самую острую проблему в обслуживании программного продукта.

Но применение средств ALM помогает исправить данную проблему и повысить эффективность использования программного обеспечения в бизнесе.

## Роли участников в системе

Идеальная среда ALM позволяет интегрировать всех участников жизненного цикла приложения, обеспечить им согласованный доступ к соответствующим ресурсам и задачам и при этом понимать контекст работы каждой отдельной роли, предоставляя ее исполнителям нужные инструменты [5].

К основным участникам процессов системы ALM можно отнести:

* топ-менеджеры – управляют проектами и контролируют ключевые параметры жизненного цикла ПО;
* менеджеры проектов – создают план по выполнению проекта и занимаются его контролем, проводят анализ рисков и организуют распределение ресурсов;
* аналитики – осуществляют связь с пользователями, формируют требования к конечному продукту и управляют ими на протяжении всего жизненного цикла;
* архитекторы – занимаются архитектурой программного продукта;
* разработчики – занимаются созданием программного обеспечения путем программирования в различных средах разработки и используют разнообразные инструменты для отслеживания качества ПО на этапе кодирования;
* тестировщики – занимаются тестированием программного продукта путем создания тестов различного плана, также выполняют тестирование производительности приложения;
* операционный персонал – отвечает за мониторинг программного обеспечения, и осуществляют обратную связь с разработчиками по поводу возникающих проблем;
* менеджеры выпуска - занимаются выпуском готового решения и его сертификацией для поставки и развертывания [19].
* пользователи – высказывают требования для проекта с помощью специализированных средств, сообщают о возникающих проблемах и ошибках в программном продукте и следят за вносимыми изменениями.

## Отличие систем ALM от систем SDLC

Как уже упоминалось ранее, очень часто управление жизненным циклом приложений (ALM) напрямую сопоставляют с жизненным циклом разработки программного обеспечения (SDLC), т.к. и тот, и другой направлены на разработку программного обеспечения. Но ALM - это гораздо больше, чем SDLC, и вот почему.

Жизненный цикл разработки программного обеспечения (Software Development LifeCycle) - это процесс, состоящий из нескольких этапов, включающий все действия, связанные с разработкой программного обеспечения [6]. Такой процесс не охватывает полного перечня задач, связанных с управлением всеми этапами жизненного цикла приложения.

ALM же в свою очередь, как раз распространяется на весь жизненный цикл приложений. Управление жизненным циклом приложений - это интеграция людей, процессов и инструментов, которые составляют важную часть в жизненном цикле приложения, начиная от момента возникновения идеи о создании программного продукта и до полного вывода продукта с рынка потребления [7]. Управление жизненным циклом приложений включает в себя анализ, управление требованиями, разработку, проектирование, тестирование, ввод в эксплуатацию и, наконец, вывод из эксплуатации.

Для упрощения можно сказать, что ALM - это расширенный процесс по управлению жизненным циклом приложения, а SDLC является частью процесса ALM, который может появляться в жизненном цикле при разработке программного обеспечения.

## Преимущества в применении систем ALM

Система ALM стал очень важной частью для многих организаций, занимающихся разработкой программного обеспечения. И все это благодаря множеству преимуществ, которые она предлагает:

* ALM синхронизирует все этапы жизненного цикла разработки приложения, а не просто фокусируется на технической стороне этапа разработки приложений.
* ALM обеспечивает координацию всех участников разработки, которая позволяет предоставлять приложения такого уровня, чтобы они отвечали потребностям бизнеса.
* ALM обещает возможность снижения затрат в организации, где оно используется надлежащим образом.
* ALM увеличивает темпы продаж для предприятий и снижает затраты на процессы в организациях.
* ALM «повышает гибкость, сокращая время, необходимое для создания и адаптации приложений, поддерживающих новые бизнес-инициативы» [8].
* ALM позволяет заинтересованным сторонам принимать активное участие в разработке программного обеспечения, поскольку у них есть доступ к наблюдению и отслеживанию процессов ALM в организации.
* ALM помогает организовать грамотный и плавный обмен информацией между участниками.
* Сегодняшний интегрированный ALM, по сравнению с другими практиками, «может лучше взаимодействовать, позволяя командам, отдельным лицам и организациям лучше понимать процесс разработки приложений» [10].
* Улучшает управление проектом в организации.
* Повышает общее качество продукта.
* ALM «повышает производительность, так как команда делится лучшими практиками для разработки и развертывания, а разработчикам нужно сосредоточиться только на текущих бизнес-требованиях» [8].
* «Внедрение Application Lifecycle Management может сократить количество ошибок в жизненном цикле разработки продукта и предоставить возможности для применения правил бизнес-процессов» [11].
* Успешная интеграция инструментов ALM позволяет быстрее разрабатывать.
* ALM помогает сократить «время обслуживания путем синхронизации приложения и дизайна» [8].
* Экономя финансовые ресурсы на управленческие вопросы, ALM позволяет инвестировать эти ресурсы в такие вопросы, как человеческая компетентность и современные технологии.

## Недостатки в применении систем ALM

Несмотря на все преимущества, в ALM также существует несколько важных недостатков:

• Большинство решений ALM были созданы как дополнения, без изначально запланированного проектирования. И доминирующей структурой современных решений для управления жизненным циклом приложений является интеграция инструментов в инструмент, и эта интеграция никогда не была устойчивый, как об этом говорят разработчики, особенно когда речь идет об интеграции инструментов разных производителей [12].

• Интеграция инструментов индивидуальна для каждого проекта и является наиболее частой проблемой. Часто происходит ситуация, когда один инструмент изменяется / модифицируется, другие инструменты нуждаются в модификации.

• Иногда присутствует избыточное количество функций, что мешает в организации жизненных циклов разработки приложения.

• Инструменты ALM бывают очень сложны и дороги в обновлении.

• Инструменты ALM на основе ролей могут привести к низкой производительности и снижению эффективности.

• Инструменты жизненного цикла «обладают впечатляющим количеством избыточных и обычно несовместимых функций в таких областях, как рабочий процесс, совместная работа, отчетность, аналитика и безопасность» [12].

• Часто возникают проблемы сотрудничества между разработчиком ALM и покупателем на основе культурных и языковых проблем.

• При использовании инструментов ALM следить за отчетностью по дисциплинам бывает чрезвычайно сложно [13].

• Данные могут храниться в неструктурированном виде при попытке создать и поддерживать синхронизацию при разработке.

# Программные решения в области ALM

В данный момент на рынке программного обеспечения можно найти большое количество различных ALM продуктов от различных разработчиков. Каждый из них обладает своим набором инструментария, средств и технологий для развертывания такой платформы как управление жизненным циклом приложения.

В работе будут описаны те системы, которые в данный момент набрали наибольшую популярность среди компаний, который занимаются созданием различного программного обеспечения.

## Microsoft Visual Studio Team System

Microsoft VS Team System - это полноценное решение для управления жизненным циклом приложений, в состав которого входят различные программные средства и процессы, которые помогают в совместной работе всех участников проекта разработки.

Microsoft Visual Studio Team System является наиболее полной на сегодняшний день реализацией ALM. Обычно решения ALM других компаний создаются из различных программных инструментов с хрупкими линиями взаимодействия. А Microsoft VS Team System была спроектирована с нуля как система ALM и обеспечивает большую часть необходимой функциональности для успешного поддержания процесса управления жизненным циклом приложения. Можно сказать, что Microsoft VS Team System - это общий репозиторий для всех видов разработки, который позволяет автоматизировать процесс управления жизненным циклом и поддерживать взаимодействие различных участников, задействованных в разработке продуктов.

Архитектура Microsoft VS Team System состоит из трех основных уровней: клиент, приложение и уровень данных, работающий вместе с базой данных SQL-сервера для предоставления центрального хранилища информации.

Microsoft VS Team System состоит из сервера и набора клиентских приложений (см. рисунок 3.1).

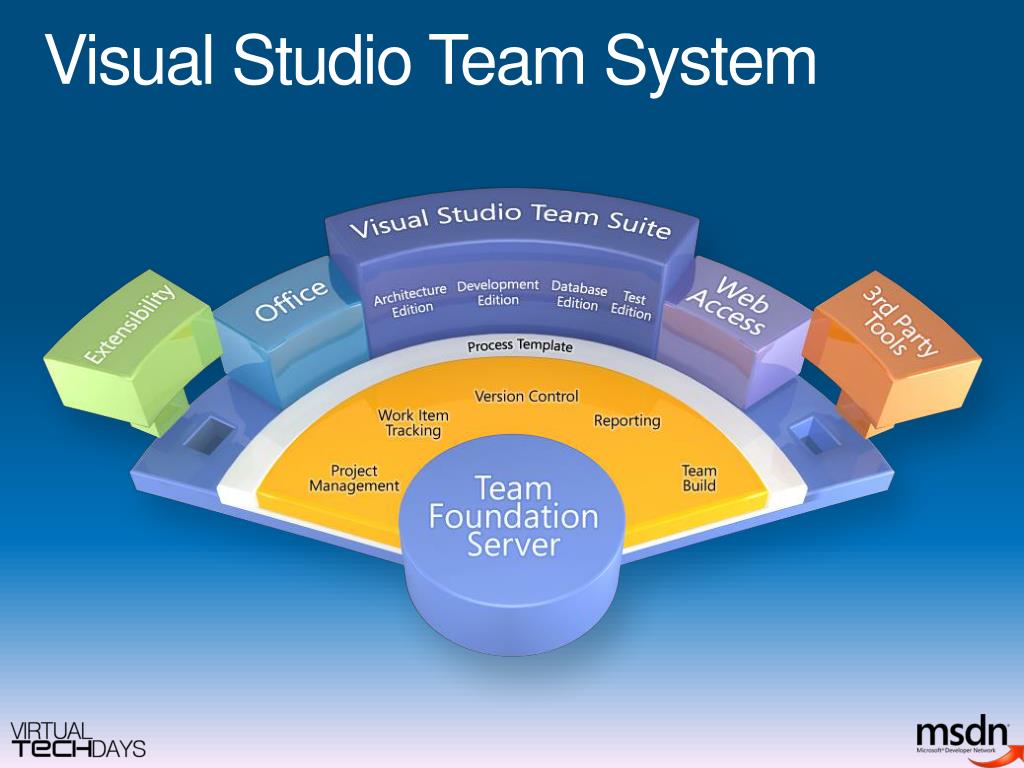


Рисунок 3.1 - Состав Visual Studio Team System

**Microsoft Visual Studio Team System - Team Foundation Server** – это ядро системы, обеспечивающее эффективную совместную работу всех членов группы и высокое качество программного обеспечения [14]. Оно соединяет функции создания отчетов, анализ проекта, система контроля версий управления проектами и сборками, управление процессами.

**Microsoft Visual Studio Team Lab Management** – позволяет конфигурировать и управлять виртуальной средой разработки и тестирования [14]. При использовании с данным инструментом Microsoft Test and Lab Manager разработчики могут обнаруживать ошибки на уровне приложений и сразу же переходить на то место, где ошибка была обнаружена.

**Microsoft Visual Studio Team System - Team Suite** предоставляет членам группы с различной специализацией интегрированный набор инструментов для создания архитектуры, проектирования, разработки и тестирования приложений и баз данных [14]. Обеспечивает совместную работу всех участников проекта и предлагает полный набор средств для каждого этапа жизненного цикла приложения.

Team Suite предлагает функциональные возможности следующих продуктов:

**Microsoft Visual Studio Team System - Architecture Edition** предназначен для совершенствования архитектуры и проверки распределенных систем [14].

**Microsoft Visual Studio Team System - Development Edition** – дает возможность разработчикам создавать безопасный и высококачественный код с минимальным количеством ошибок, которые связаны с безопасностью, и снизить вероятность возникновения ошибок на других этапах жизненного цикла.

**Microsoft Visual Studio Team System - Test Edition** – предоставляет полный набор средств тестирования приложений и веб-служб, интегрированный в среду Visual Studio [14].

**Visual Studio Team** **System** - **Database Edition** - дает возможность разрабатывать базы данных и проводить тесты на наличие ошибок.

## TeamForge

TeamForge - это система управления жизненным циклом приложений, которой поддерживается контроль версий и система управления разработкой программного обеспечения.

В интерфейсе TeamForge есть целый ряда сервисов для различных этапов жизненного цикла разработки программного обеспечения и из можно объединять с бесплатными программными приложения с открытым исходным кодом (например, PostgreSQL и Subversion).

У TeamForge есть возможность:

* Управлять документацией;
* Создавать доску задач;
* Работать с кодом приложения из хранилища;
* Обеспечивать доступ в проекте на основе ролей;
* Создавать шаблоны проекта и работать с ними.

## Devprom ALM

Devprom ALM - мощный инструмент для управления полным жизненным циклом разработки программного обеспечения, от исходной идеи до поддержки работающего продукта [15].

Он является хорошим решение для совместной работы всех участников над проектом.

Фактически всю систему Devprom ALM можно разбить на следующие функциональные блоки [15]:

1. Управление проектами и ресурсами на уровне компании.

* Поддержка современных методологий разработки.
* Возможность создавать планы по релизу продукта и отслеживать состояние проекта, а также организовывать доску задач и доску историй изменений.
* Можно создать базу знаний с необходимой терминологией для проекта.
* Систему можно объединять с другими инструментами разработки.
* Возможность создания планов итераций по разработке.
* Есть возможность производить учет времени, затраченного на каждый этап проекта.

1. Разработка и управление требованиями.

* Создавать требования можно совместно из браузера.
* Есть возможность вести обсуждения по проекту и согласовывать требования в команде.
* Документирование UML-моделей, формул и алгоритмов, просмотр изменений в моделях и формулах.
* Быстрая и простая постановка задач разработчикам и тестировщикам.
* Можно загружать и выгружать требования в разных форматах (doc, docx, html, odt и др.).
* Можно собирать информация для анализа проблем и повышения производительности.

1. Организация поддержки программных продуктов.

* Можно регистрировать клиентов автоматически.
* Регистрация заявок через браузер, на email или вручную, настройка жизненного цикла заявок.
* Можно контролировать сроки по решению запросов.
* Запросы можно переносить между командами, которые участвуют в разработке, изменять статус запросов.
* Обсуждение заявок между клиентами и сотрудниками отдела поддержки.
* Поддерживается русский и английский интерфейс системы.

1. Управление разработкой и поставкой программного кода[15].

* Интеграция с системами контроля версий Git, Github, Gitlab. Подключение неограниченного количества репозиториев.
* Автоматическое связывание коммитов с историями, доработками и требованиями.
* Лента изменений в коде репозиториев.
* Автоматическая привязка выполненных историй и пожеланий к сборкам и версиям продукта на основе коммитов.

1. Подготовка тестовой документации и управление тестированием [15].

* Можно разрабатывать планы тестов на основе требований и пользовательских историй.
* Поддержка ручного и автоматического тестирования для нескольких версий продукта.
* Поддержка тестовой документации в актуальном состоянии.
* Полностью настраиваемый процесс работы над тестовой документацией.
* Отчеты и метрики для контроля качества продуктов и процесса тестирования.

1. Разработка эксплуатационной документацией.

* Быстрое создание через браузер файлов справок, рабочих инструкций, руководств пользователя, контекстной помощи.
* Можно загружать и выгружать документы в различных форматах.
* Есть возможность создавать документацию для пользователей.
* Процесс создания технической документации можно контролировать и настраивать.
* Можно обсуждать документацию при участии всех команд, участвующих в разработке.

## Kovair ALM Studio

Kovair ALM Studio - это платформа, которая дает возможность работать совместно всем участника проекта на всех этапах управления жизненным циклом приложения.

Благодаря тому, что в системе есть отслеживаемость всех этапов проекта и можно создавать отчеты в режиме реального времени с использованием данных со всех этапов жизненного цикла, Kovair ALM Studio предоставляет единый центр информации для всех пользователей, участвующих в каждом из этапов жизненного цикла.

В Kovair ALM можно организовать рабочий процесс на основе задач. Это позволит организациям внедрять и применять централизованное управление на всех этапах жизненного цикла проекта.

Данный механизм исключит передачу информации между командами вручную, снизит затраты, повысит автоматизацию процессов разработки программного обеспечения, а также повысит качество продукции.

К списку основных возможностей можно отнести:

* Легкая настройка конфигурации для проекта;
* Прослеживаемость проекта на всех этапах разработки;
* Организация рабочего процесса на основе задач и требований;
* Создание отчетов в режиме реального времени и мониторинг процессов в проекте;
* Возможность интегрировать в систему другие инструменты.

## SwiftALM

SwiftALM является совместной платформой для управления ИТ-программами, проектами и процессами.

SwiftALM предлагает уникальное сочетание возможностей:

* управления проектами и программами,
* управления каждым этапом жизненного цикла разработки программного обеспечения,
* управление процессами и совместной работой [16].

SwiftALM помогает организациям по разработке программного обеспечения сориентироваться на потребности рынка.

Используя сложные функции моделирования организации и контроля доступа в SwiftALM, руководители различных уровней организации могут получить соответствующие сводные представления программ, проектов и продуктов с возможностью выполнения нескольких уровней детализации информации [16].

SwiftALM позволяет создавать шаблоны проектов для последующего их использования в будущих проектах. Также создается структура, которая помогает правильно и оперативно спланировать все процессы разработки приложения, и при этом предоставит каждому участнику на основе его роли в проекте свой набор инструментов.

Управление процессами SwiftALM позволяет моделировать процессы предприятия и отслеживать организационные показатели и цели для проектов, программ и продуктов [16].

## Open ALM

Open ALM от компании Borland - мощный инструмент для управления всеми этапами жизненного цикла приложения.

Open ALM обеспечивает функциональную отдачу от ALM и одновременно предоставляет непревзойденную степень гибкости на уровнях процессов, инструментальных средств и платформ [17].

Платформа Open ALM и поддерживающие ее средства предоставят возможности в виде:

* Настройка метрик прогресса, просчет рисков и качества проекта, можно поддерживать управление разработкой продукта.
* Полный оперативный контроль и отслеживание жизненного цикла. Это позволит добиться реального согласования бизнес-целей и действий в процессе разработки, обеспечит лучшую связь между ожиданиями конечных пользователей и итогами проекта [17].
* Согласованная работа специалистов и средств на основе процессов.

Эти возможности обеспечивают прекрасную эффективность работы коллектива, поддерживают инициативы по улучшению качества и облегчают бремя необходимости соответствия внутренним и внешним нормам. Они будут предоставляться в виде набора компонентов инфраструктурного уровня и корпоративных средств управления ALM [17].

Инструментальные средства от Borland для разработки приложений и управления проектами позволит получить отдачу в четырех основных областях процессов:

* Контроль разработки ПО;
* Возможность управлять требованиями и создавать их;
* Управление качеством продукта;
* Управление изменениями.

## SAP Solution Manager

SAP Solution Manager - это продукт от компании SAP AG, интегрированная платформа, которая способствует повышению эффективности ALM при ее интеграции с другими инструментами [18]. SAP охватывает несколько видов деятельности ALM.

Платформа SAP Solution Manager является очень важной частью, которая обеспечивает функциональность ALM. SAP Solution Manager организует доступ ко всей необходимой информации проекта, а также задачам, инструментам и методам. Чтобы операции ALM можно было выполнять быстро и с минимальными затратами, необходимо правильно организовать работу диспетчера решений SAP на протяжении всего жизненного цикла продукта.

Работу SAP Solution Manager можно разделить на несколько этапов:

1. Создание плана проекта;
2. Разработка и тестирование;
3. Реализация и поддержка конечного продукта.

Первый этап, называется планом проекта. Он включает в себя выбор проекта, для которого установлен инструментарий SAP Solution Manager. Сценарий либо выбирается из репозитория, либо составляется командой разработки. После настройки проекта SAP Solution Manager предоставляет службы для управления конфигурациями и их реализации.

Следующие этапы - это разработка и дальнейшее тестирование, которая осуществляется с помощью SAP Solution Manager.

Реализация проекта включает в себя следующие моменты:

• Постоянное слежение за выполнением целей проекта, особенно важно следить за затратами на проект, сроками выполнения и объемом потраченных ресурсов;

• Оперативно управлять изменениями на всех этапах проекта;

• Разработка учебных материалов и документации для конечных пользователей.

• Создание ролей для пользователей и реализация моментов, связанных с безопасностью и авторизацией.

• Внедрение спецификаций реализации для разработок заказчика.

Клиентам, использующим SAP Solution Manager, оказывается активная поддержка по внедрения системы в свой проект, а также осуществляется техническая поддержка.

Также для облегчения процесса внедрения системы в проект в системе предусмотрен "модуль электронного обучения", который является обязательной частью диспетчера решений и предназначен для оказания поддержки клиентам с помощью учебных материалов.

# Применение системы ALM для управления разработкой приложения

## Выбор системы ALM

В качестве платформы ALM на ее применение в отделе Управления информационными технологиями была выбрана система Devprom ALM. Выбор данной системы из ранее показанных в работе был произведен по нескольким причинам:

* Разработкой данной системы занимается российская компания;
* Полноценный русский интерфейс;
* Грамотная и оперативная служба поддержки, а также есть возможность записаться на индивидуальные консультации по вопросам о данной платформе;
* Система разрабатывалась на основе опыта, полученного от российских компаний разработчиков, что тоже плюс.
* Стоимость владения данной системой на порядок ниже по сравнению с зарубежными аналогами.
* Можно решить, какие модули системы стоит покупать для работы над проектом.

На сегодняшний день платформу Devprom ALM можно считать самым оптимальным решением для российских разработчиков программного обеспечения за счет:

* гибкой настройки системы;
* внедрения системы в минимальные сроки;
* полного охвата всех этапов жизненного цикла разработки программного обеспечения;
* невысокой стоимости владения.

## Исходные данные

Перед тем как перейти к работе с демонстрационной версией платформы Devprom ALM и создать на ней проект, необходимо дать описание приложению, на основе которого будет создаваться проект в системе Devprom ALM.

В качестве приложения была взята программа для перекодировки текста. Программа написана на языке C# в среде разработки программного обеспечения Visual Studio 2017. Изначально программа разрабатывалась без использования систем ALM. Поэтому далее будет показано, как бы могла происходить работа над данным приложением уже с использованием системы ALM.

Интерфейс приложения показан на рисунке 4.1

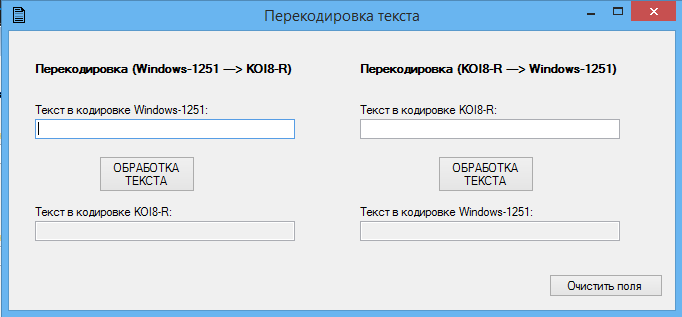


Рисунок 4.1 - Интерфейс приложения

С листингом программы можно ознакомиться ниже:

using System;

using System.Text;

using System.Windows.Forms;

namespace ProgramLab3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text != "")

{

textBox2.Clear();

string t = textBox1.Text;

Encoding koi8r = Encoding.GetEncoding("koi8r");

Encoding win1251 = Encoding.GetEncoding("Windows-1251");

byte[] koi8rBytes = win1251.GetBytes(t);

byte[] win1251Bytes = Encoding.Convert(koi8r, win1251, koi8rBytes);

textBox2.Text = win1251.GetString(win1251Bytes);

}

else

{

MessageBox.Show("Заполните поля!!!", "Сообщение", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button2\_Click(object sender, EventArgs e)

{

if (textBox3.Text != "")

{

textBox4.Clear();

string t = textBox3.Text;

Encoding koi8r = Encoding.GetEncoding("koi8r");

Encoding win1251 = Encoding.GetEncoding("Windows-1251");

byte[] win1251Bytes = koi8r.GetBytes(t);

byte[] koi8rBytes = Encoding.Convert(win1251, koi8r, win1251Bytes);

textBox4.Text = koi8r.GetString(koi8rBytes);

}

else

{

MessageBox.Show("Заполните поля!!!", "Сообщение", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button3\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

}

}

}

## Создание проекта и описание доступных функций платформы Devprom ALM

Для ознакомления с возможностями платформы Devprom по работе с проектами, на официальном сайте программы https://www.devprom.ru [15] имеется 2 варианта по работе с пробной версией программы сроком на 30 дней:

* развертывание системы в "облаке"
* установка программы из дистрибутива, который можно скачать с официального сайта, для нужной версии операционной системы.

Было выбрано решение по работе с системой Devprom в "облаке". Для начала работы необходимо указать название сайта, на котором будет располагаться данная пробная версия системы и производиться тестирование ее возможностей, а также указываются регистрационные данные пользователя: имя и адрес электронной почты (см. рисунок 4.2).

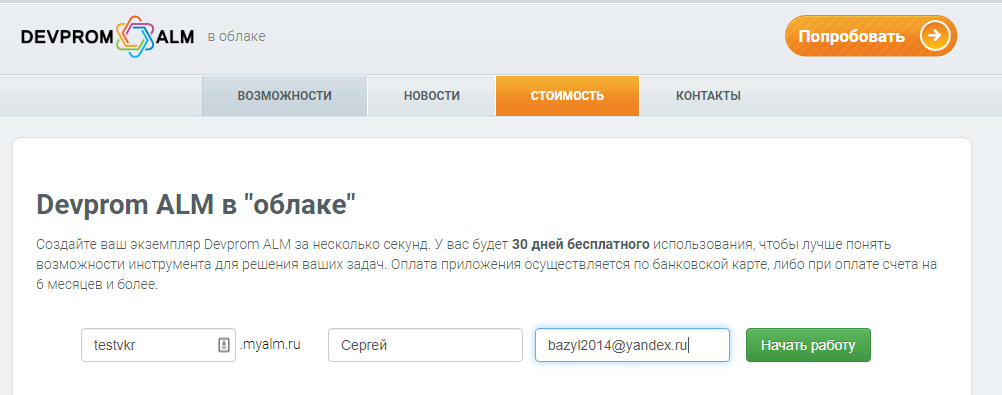


Рисунок 4.2 - Ввод регистрационных данных для работы с системой Devprom в "облаке"

Дальше происходить переход на сайт, на котором уже можно начать работу с пробной версией системы Devprom ALM (см. рисунок 4.3).

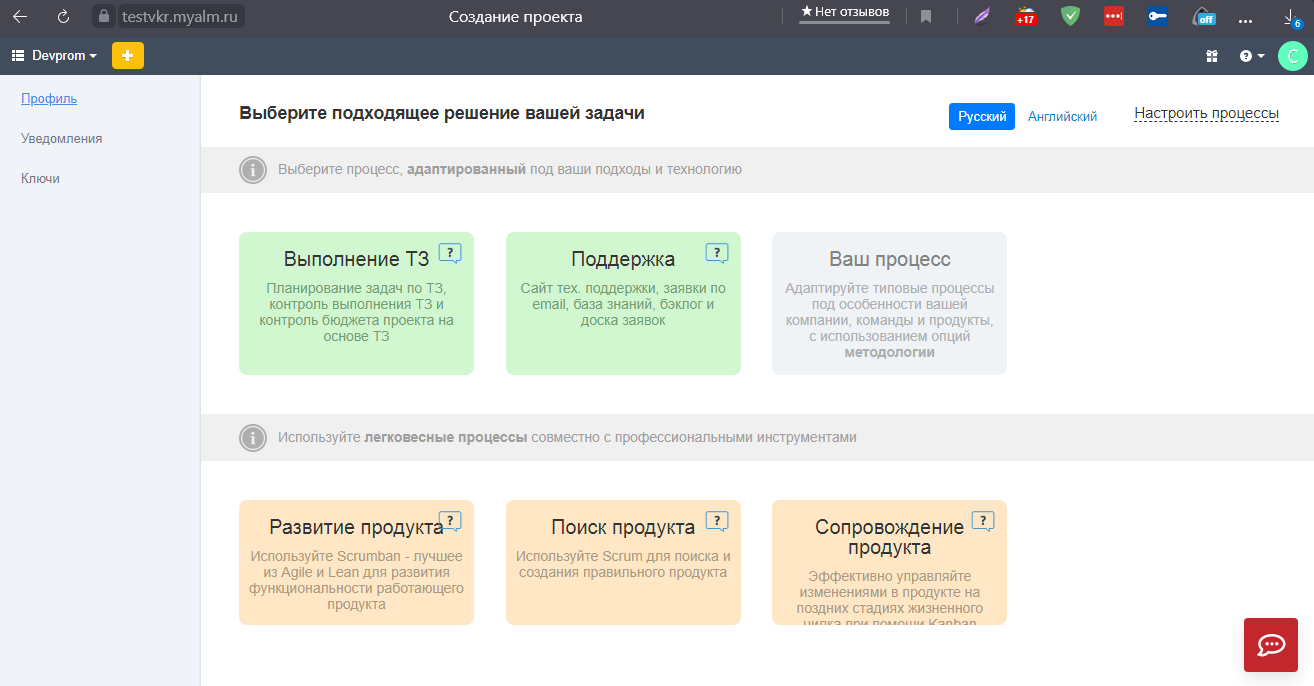


Рисунок 4.3 - Страница создания проекта

На данной странице пользователю необходимо сделать выбор в сторону того процесса, который он хотел бы использовать для решения своей задачи. На выбор представлены следующие типовые процессы, которые чаще всего используются компаниями разработчиками ПО: Выполнение ТЗ, Поддержка, Развитие продукта, Поиск продукта, Сопровождение продукта. Также есть возможность создать свой процесс, который будет адаптирован под особенности разработки конкретной компании.

Перед выбором процесса, который поможет организовать разработку приложения, хотелось бы рассказать о том, что структура разработки программного обеспечения представляет собой набор подпроцессов, выполнение каждого из которых приводит к получению готового продукта. Данную структуру можно наглядно рассмотреть на диаграмме процессов нотации IDEF0 (см. рисунок 4.4 и 4.5)

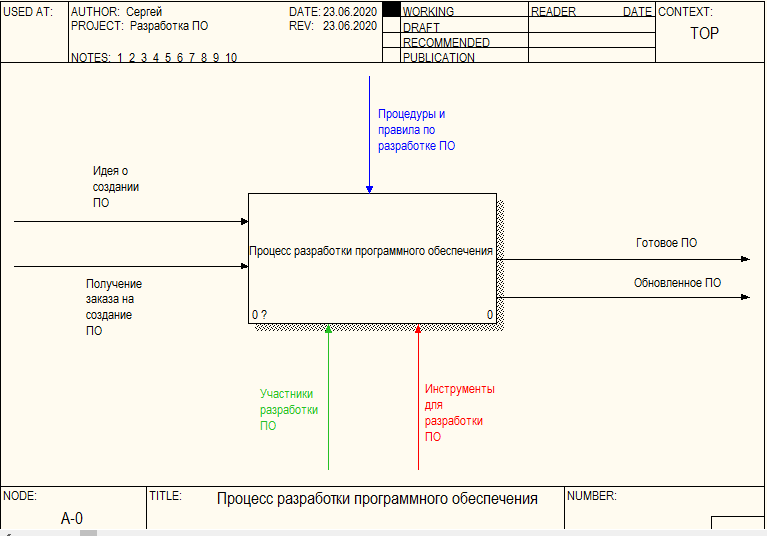


Рисунок 4.4 - Диаграмма процесс разработки ПО

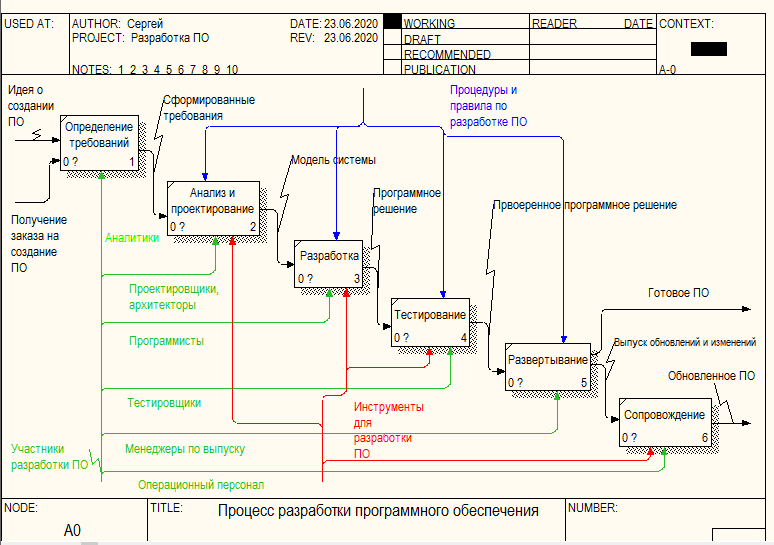


Рисунок 4.5 - Диаграмма процесс разработки ПО

Для ознакомления с возможностями программного решения Devprom ALM был выбран процесс "Выполнение ТЗ", который направлен на организацию таких подпроцессов как "разработка" и "тестирование".

Далее появляется страница, на которой необходимо дать название проекту, его кодовое название, которое будет использоваться в адресной строке для поиска данного проекта, а также можно: начать проект с чистого листа, добавить в него демонстрационные или импортировать данные из трекера (см. рисунок 4.4).

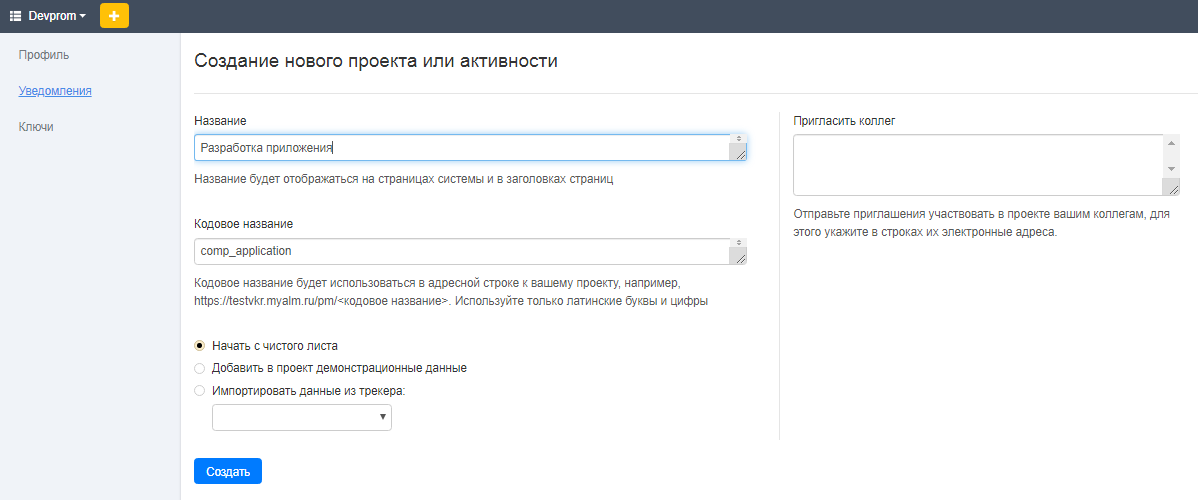


Рисунок 4.4 - Создание проекта

**Меню "Избранное"**

После создания проекта, пользователь оказывается во вкладке "Панель управления" меню "Избранное". В данной вкладке можно размещать ячейки, которые будут отражать интересуемую для разработчиков информацию по проекту. Для данного проекта панель управления выглядит следующим образом (см. рисунок 4.5).

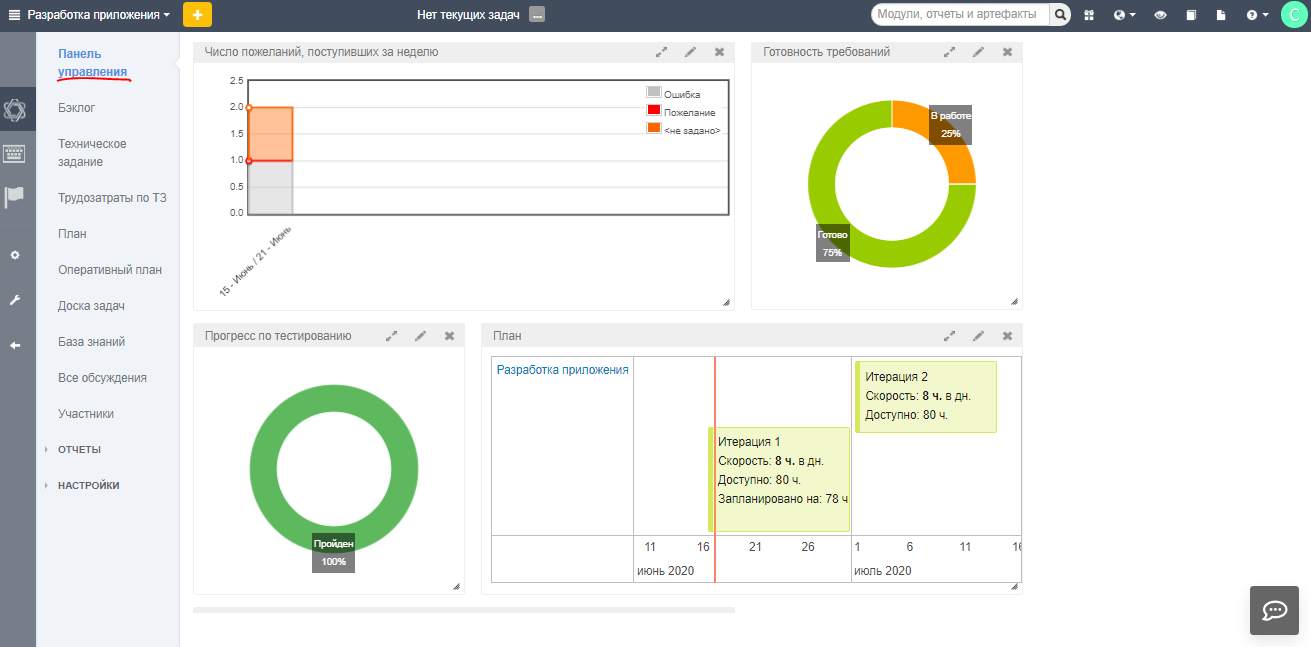


Рисунок 4.5 - Панель управления

Также в меню "Избранное" есть следующие разделы: бэклог, техническое задание, трудозатраты по ТЗ, план, оперативный план, доска задач, база знаний, все обсуждения, отчеты и настройки.

В разделе "Бэклог" можно создавать список работ, которые необходимо выполнить разработчикам. Задача может быть по устранению каких-либо ошибок в программе, либо по изменению или доработке ее функционала и т.п. Для проекта программы по перекодировки было внесено две задачи, которые можно увидеть на рисунке 4.6.

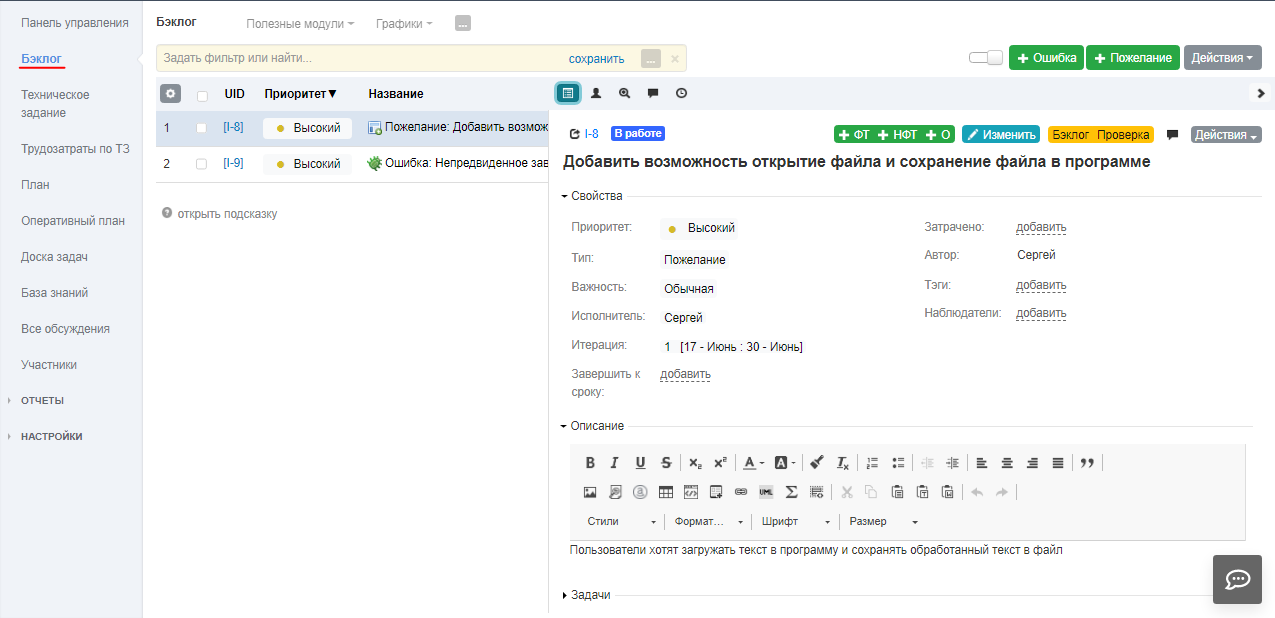


Рисунок 4.6 - Раздел "Бэклог"

Раздел "Техническое задание" содержит основную информацию о проекте, которой необходимо следовать для того, чтобы реализовать продукт такого уровня качества, который будет устраивать клиента. Пример технического задания показан на рисунке 4.7.

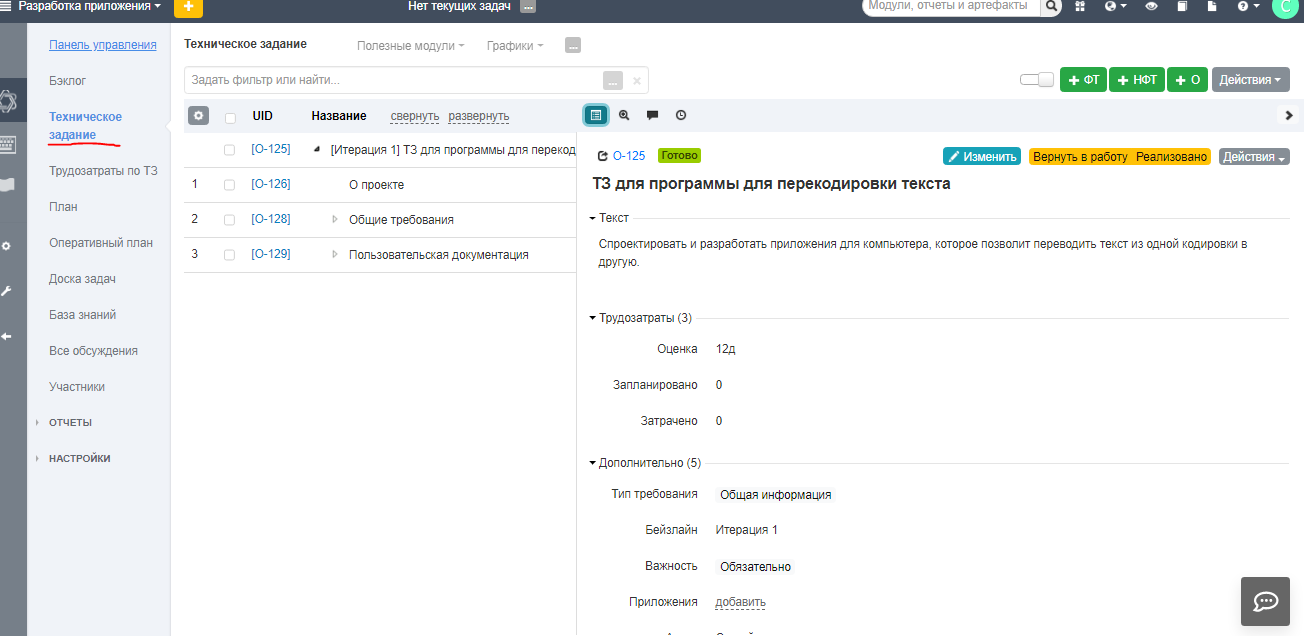


Рисунок 4.7 - Раздел "Техническое задание"

В разделе "Трудозатраты по ТЗ" можно ознакомиться с информацией об оценке трудоемкости реализации требований проекта и о кол-ве времени, фактически затраченного на реализацию (см. рисунок 4.8).

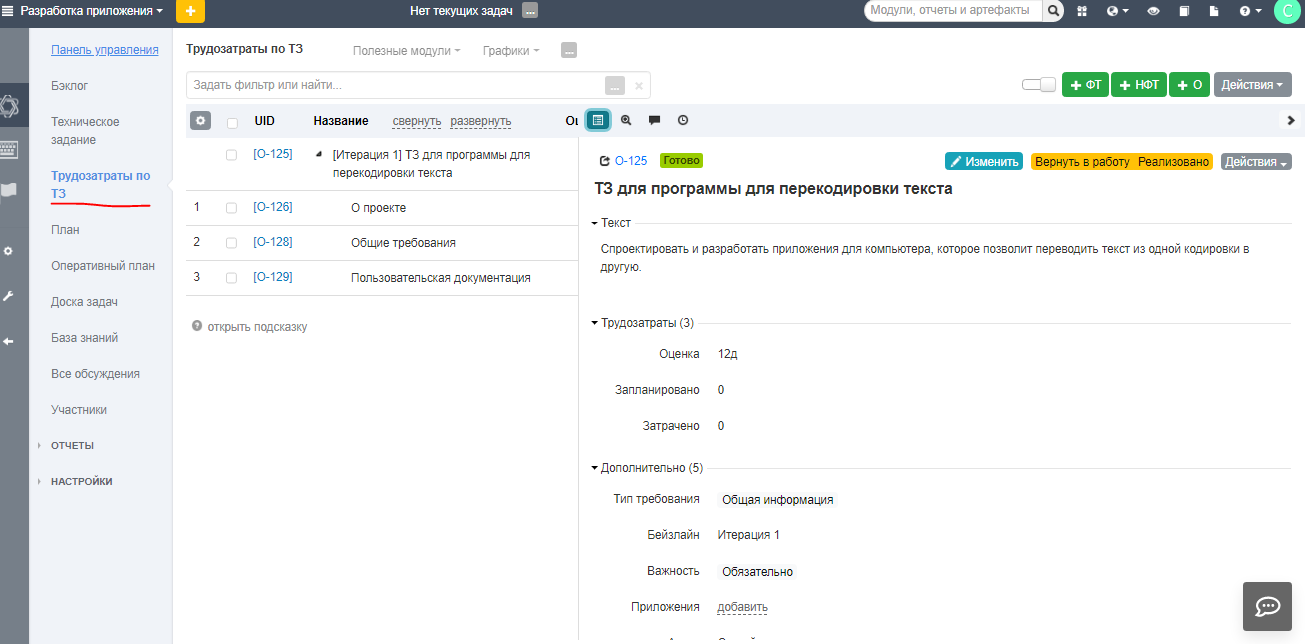
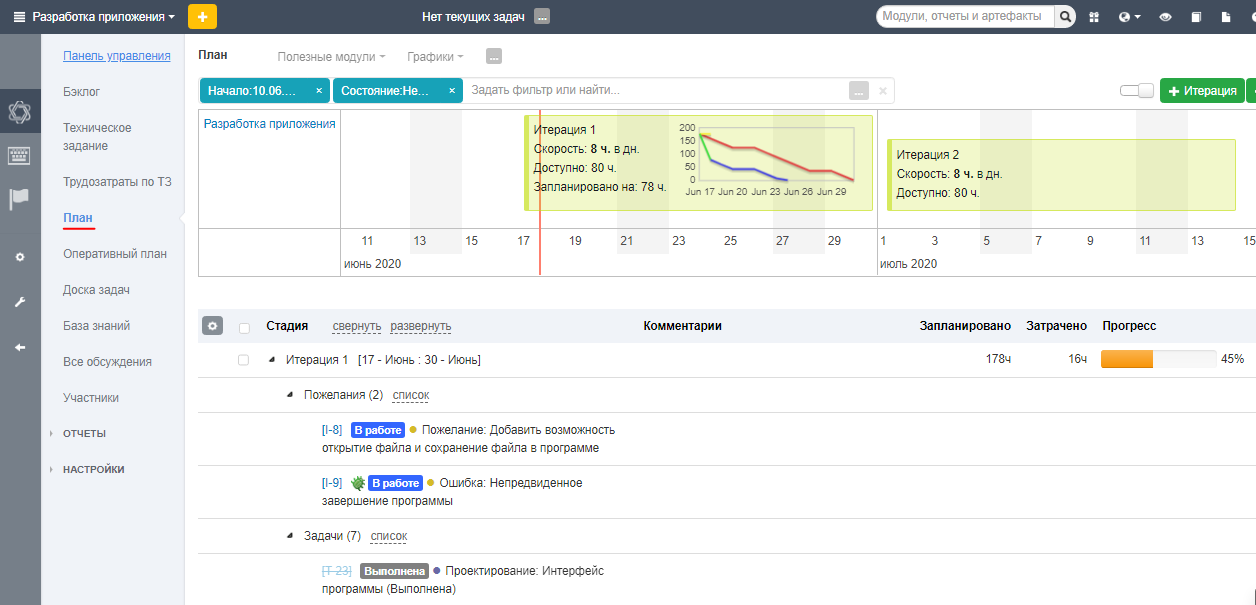


Рисунок 4.8 - Раздел "Трудозатраты по ТЗ"

Раздел "План" помогает следить за интервалами времени, в рамках которых необходимо выполнить заданный объем работы. Для проекта в данной работе план выглядит следующим образом (см. рисунок 4.9).



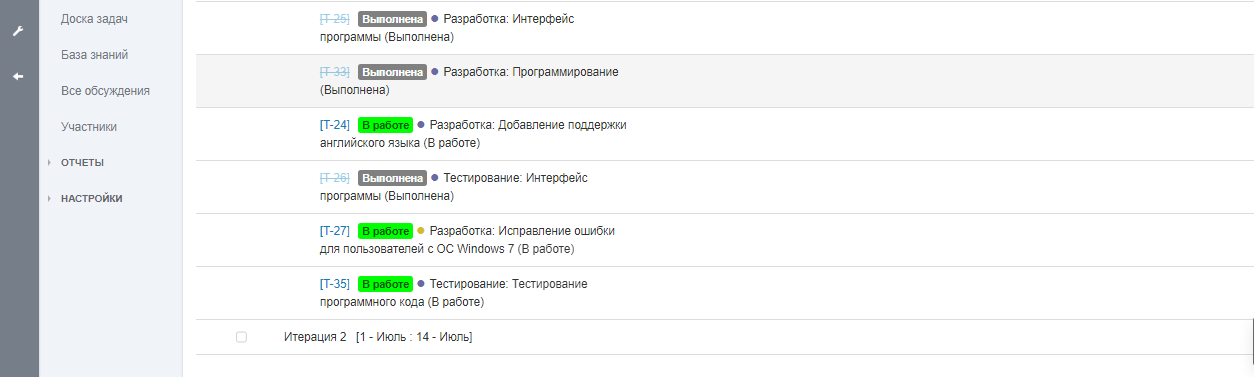


Рисунок 4.9 - Раздел "План"

Раздел "Оперативный план" необходим для контроля загрузки исполнителей проекта и объема работ, запланированных в итерациях. Это способствует эффективному перераспределению задач между итерациями и исполнителями. Пример такого плана представлен на рисунке 4.10.

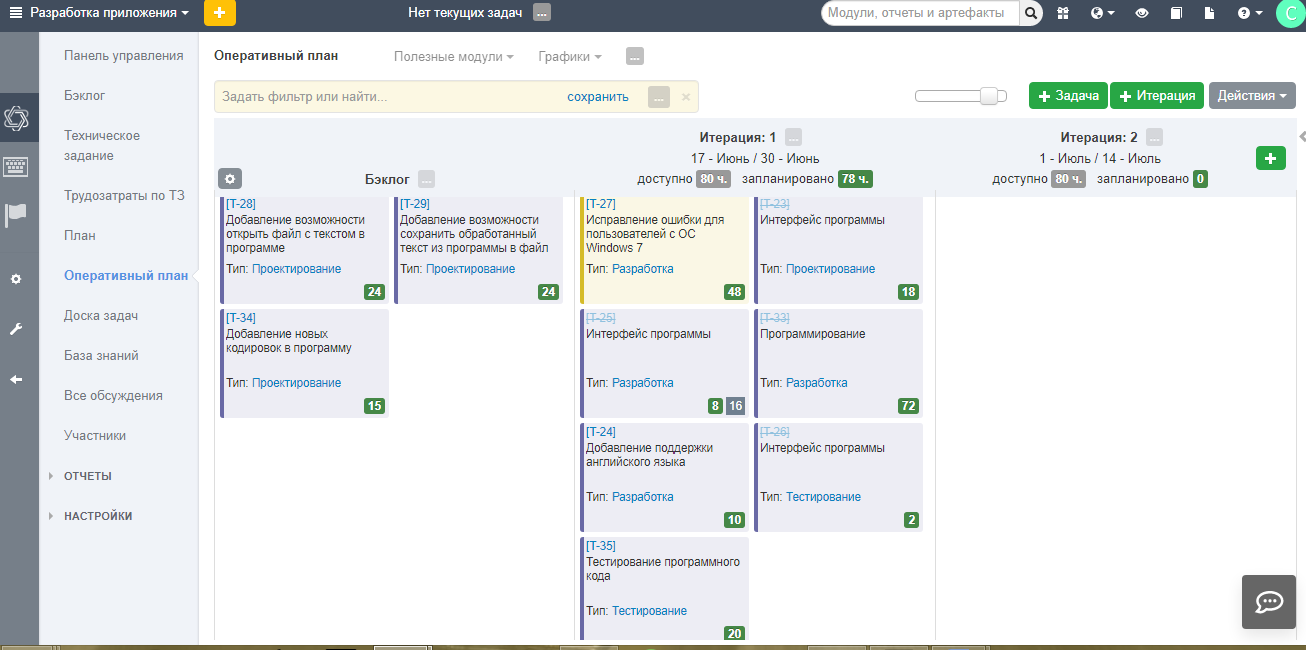


Рисунок 4.10 - Раздел "Оперативный план"

Раздел "Доска задач" предназначен для того, чтобы отслеживать различные состояния проекта разработки: какие задачи были добавлены на выполнения, какие задачи уже выполняются, а какие задачи были уже успешно завершены. Для данного проекта доска задач выглядит следующим образом (см. рисунок 4.11).

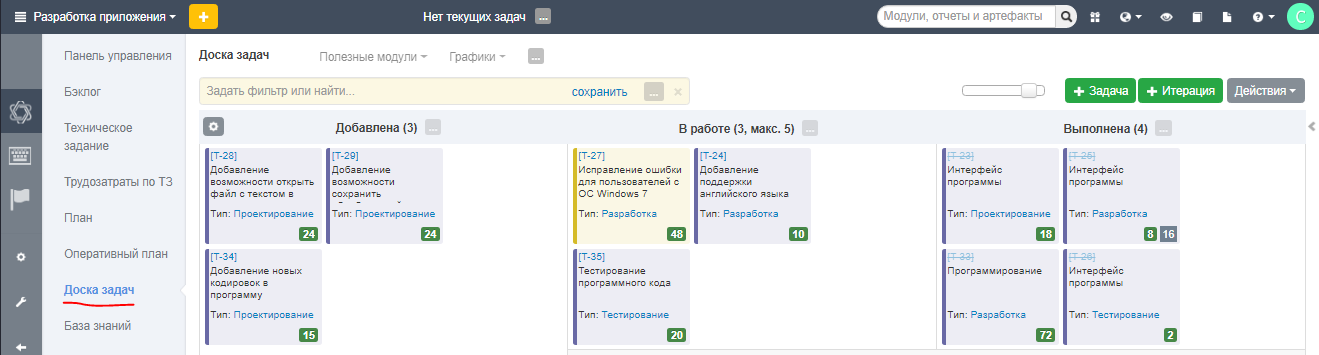


Рисунок 4.11 - Раздел "Доска задач"

Раздел «База знаний" платформы Devprom ALM поможет организовать и подготовить информацию о содержании проекта, используемой терминологии, зафиксировать особенности работ по проекту и другую необходимую информацию.

Раздел "Все обсуждения" позволяет ознакомиться с информацией об изменениях, внесенных в проект, кем были внесены изменения, какого рода были внесены изменения и т.п.).

Раздел "Участники" отображает список участников проекта, также позволяет определить, кому из пользователей можно заходить в проект, в какой роли и с какими правами доступа (см. рисунок 4.12).

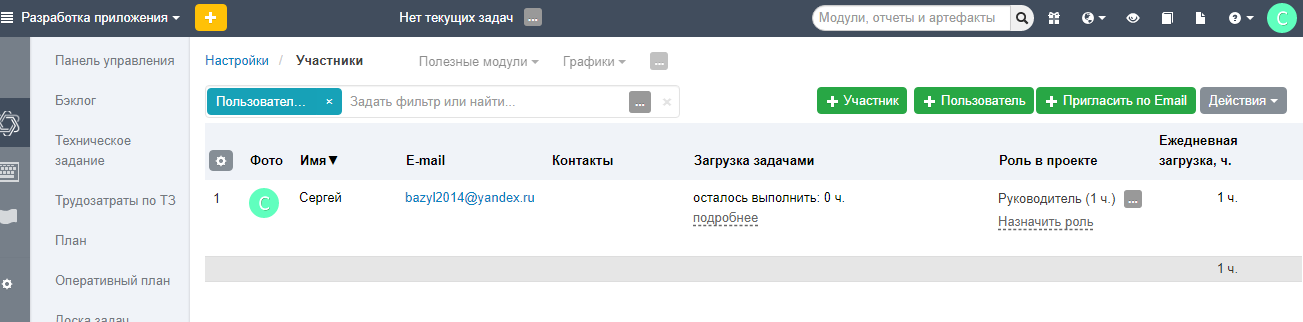


Рисунок 4.12 - раздел "Участники"

Помимо указанных разделов основного меню есть возможность ознакомиться с отчетами по графику разработки требований проекта, загрузкой участников задачами, посмотреть информацию о затраченном времени и журнал изменений.

**Меню "Разработка"**

В меню "Разработка" содержаться следующие разделы по проекту: доска задач разработки, коммиты, файлы, ревью, репозитории, тестовые отчеты, сборки и окружения.

Раздел "Доска задач разработки" помогает отслеживать именно те задачи, которые связаны непосредственно с разработкой. Данные задачи так же, как и в разделе "Доска задач", делятся на следующие задачи:

* задачи, которые были добавлены для выполнения;
* задачи, которые уже находятся в исполнении;
* и задачи, которые были успешно завершены.

Данная доска может выглядеть следующим образом (см. рисунок 4.13).

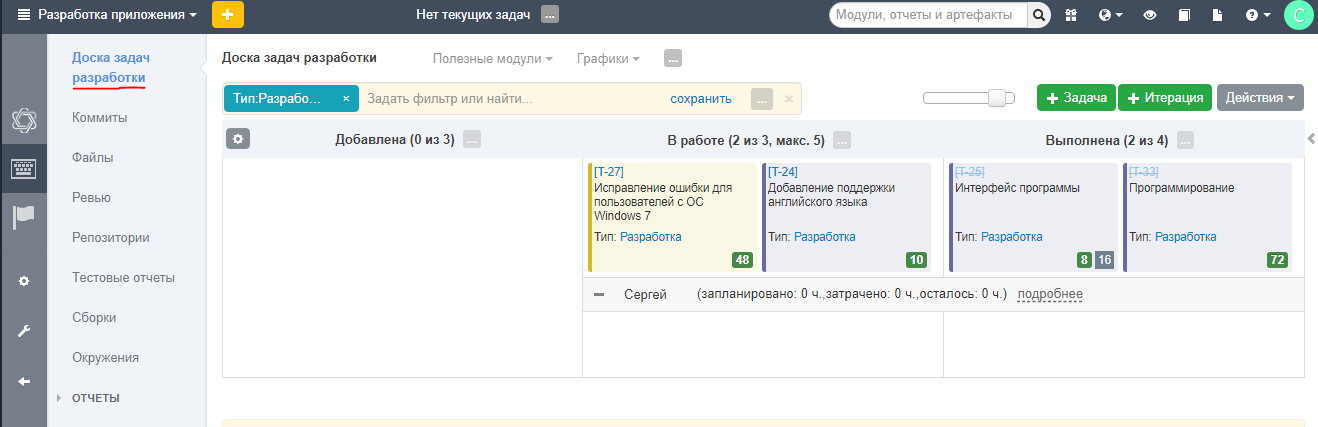
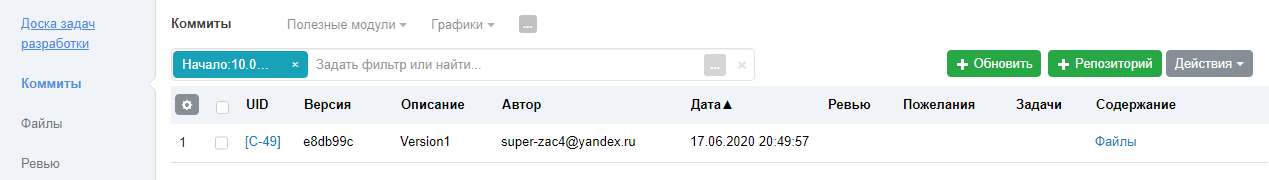


Рисунок 4.13 - Раздел "Доска задач разработки"

Раздел "Коммиты" позволяет добавить репозиторий, в котором могут находиться все необходимые файлы программы. Через данный раздел разработчики могут отслеживать изменения любых файлов, которые были внесены в заданный репозиторий. Также данный раздел позволяет в случае совершения критической ошибке при работе над кодом программы откатить разработку до того состояния, когда в программе все работало хорошо.

Для проекта, который описывается в данной работе, "коммиты" выглядят так, как показано на рисунке 4.14.



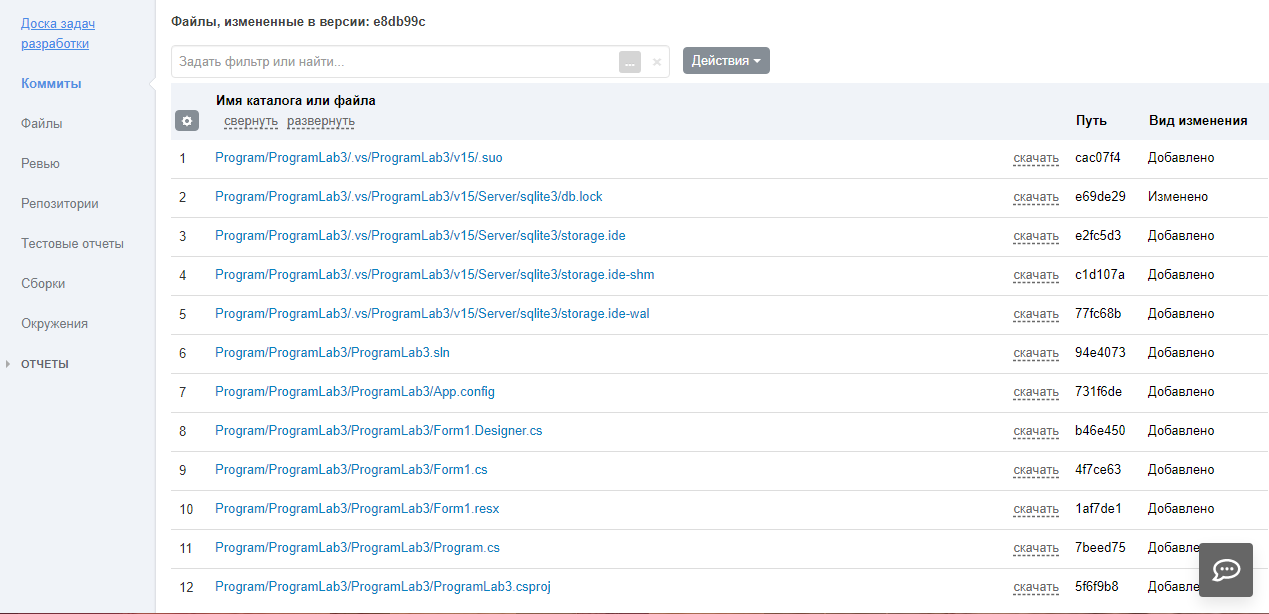


Рисунок 4.14 - Раздел "Коммиты"

Раздел "Файлы" позволяет посмотреть, какие файлы присутствует в проекте приложения (см. рисунок 4.15).

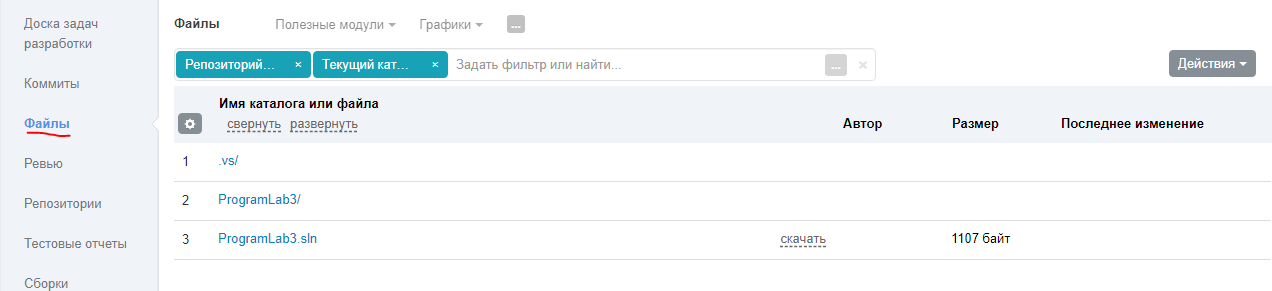


Рисунок 4.15 - Раздел "Файлы"

Раздел "Ревью" позволяет использовать из него информацию для разрешения перевода задачи или требования на следующий этап разработки приложения (см. рисунок 4.16)

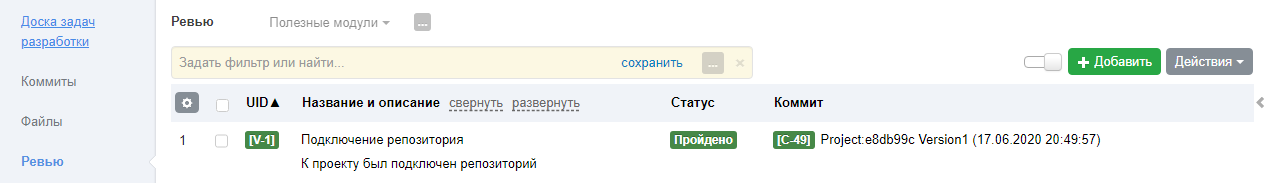


Рисунок 4.16 - Раздел "Ревью"

В разделе "Репозитории" можно узнать более подробную информацию о репозитории: путь до него, краткое название, сколько было загружено коммитов для каждого репозитория (см. рисунок 4.17).

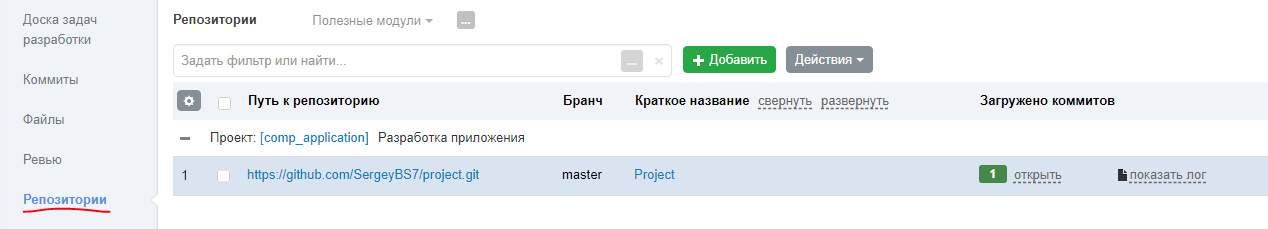


Рисунок 4.17 - Раздел "Репозитории"

В разделе "Тестовые отчеты" можно ознакомиться с информацией о проводимых тестах по проекту: для какой задачи тест проводился, какая была версия приложения, какие ошибки были обнаружены при тестировании и др.

В разделе "Сборки" отображаются все добавленные сборки приложения, их статус, информация о тестах, проводимых с приложением, реализованных требованиях и обнаруженных дефектах (см. рисунок 4.18).

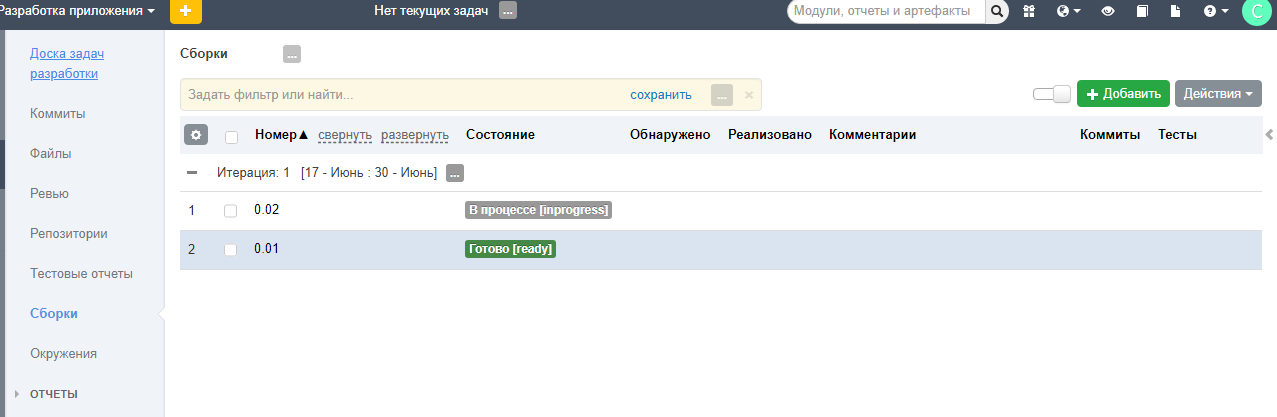


Рисунок 4.18 - Раздел "Сборки"

Раздел "Окружения" дает возможность зафиксировать и ознакомиться с информацией о том, в какой компьютерной системе развертывалось и выполнялось разрабатываемое программное обеспечение.

**Меню "Тестирование"**

Данное меню содержит в себе следующие разделы: тест - планы, дерево сценариев, доска сценариев, доска задач тестирования, тестовые сценарии, тестовые отчеты, устаревшие сценарии, сборки и окружения.

Раздел "Тест - планы" позволяет объединять тестовые сценарии по видам тестирования (функциональное, стресс-тестирование и т.п.), по назначению, по важности и позволяет выполнять тестирование очередной итерации или релиза продукта. Для каждого тест - плана показывается его состояние и его готовность (см. рисунок 4.19).

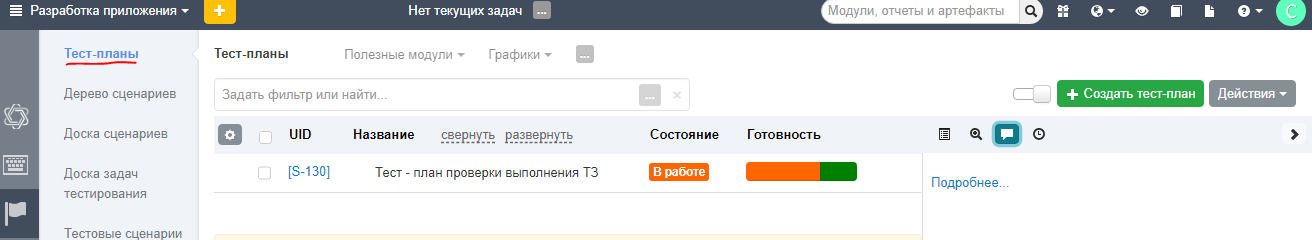


Рисунок 4.19 - раздел "Тест - планы"

Раздел "Дерево сценариев" позволяет добавить тестовый сценарий со следующий информацией: какой будет вид у теста, какой именно раздел будет тестироваться, насколько важен тест, какова его трудоемкость и др. (см. рисунок 4.20).

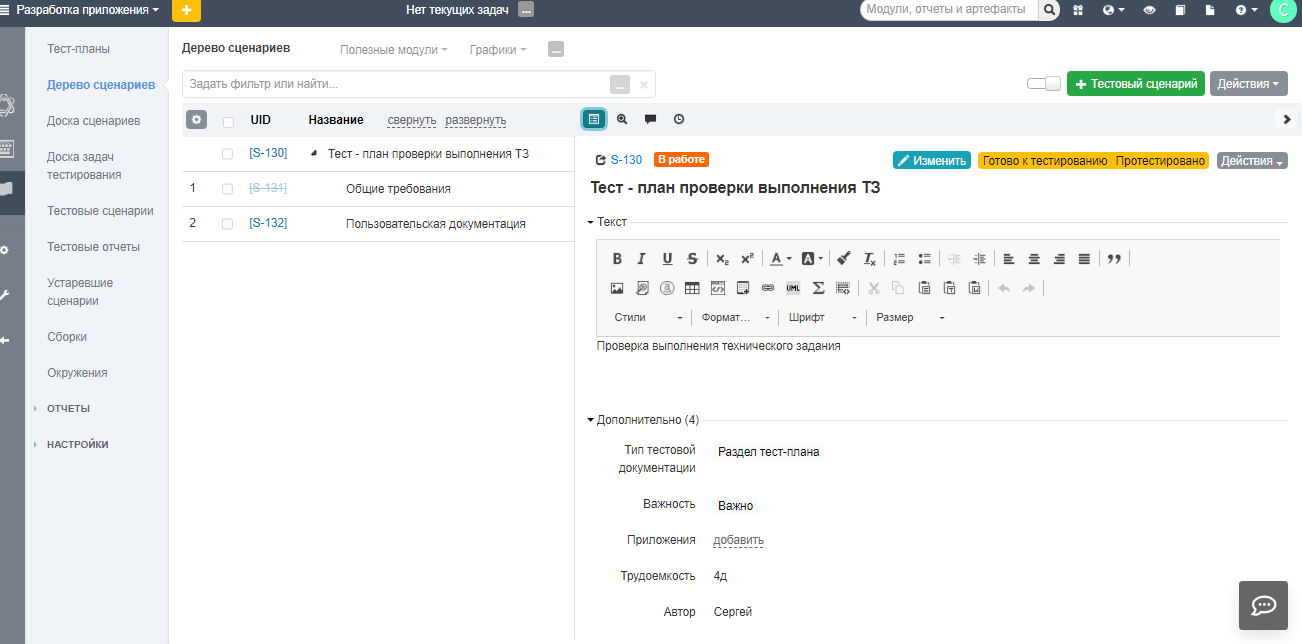


Рисунок 4.20 - Раздел "Дерево сценариев"

Раздел "Доска сценариев" позволяет управлять планом по тестированию программного обеспечения (см. рисунок 4.21)

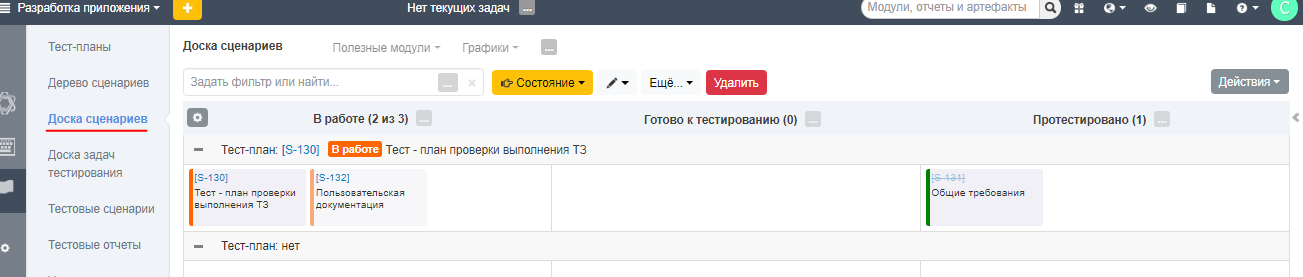


Рисунок 4.21 - Раздел "Доска сценариев"

Раздел "Доска задач тестирования" помогает организовать план по тестированию различных этапов разработки и отслеживать его. Пример такой доски показан на рисунке 4.22.

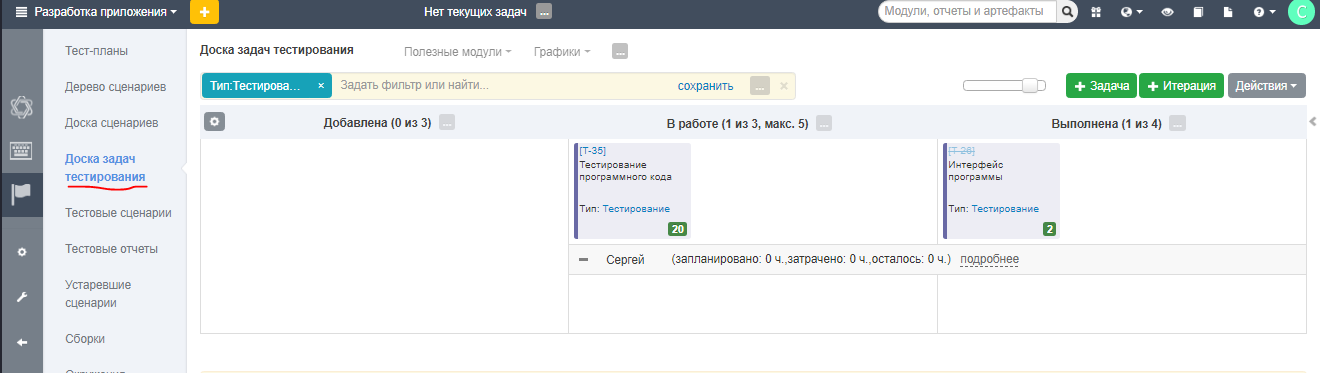


Рисунок 4.22 - Раздел "Доска задач тестирования"

Раздел "Тестовые сценарии" помогает работать со списком созданных тестовых сценариев, несмотря на то, в каком тест - плане находится тот или другой тестовый сценарий (см. рисунок 4.23)

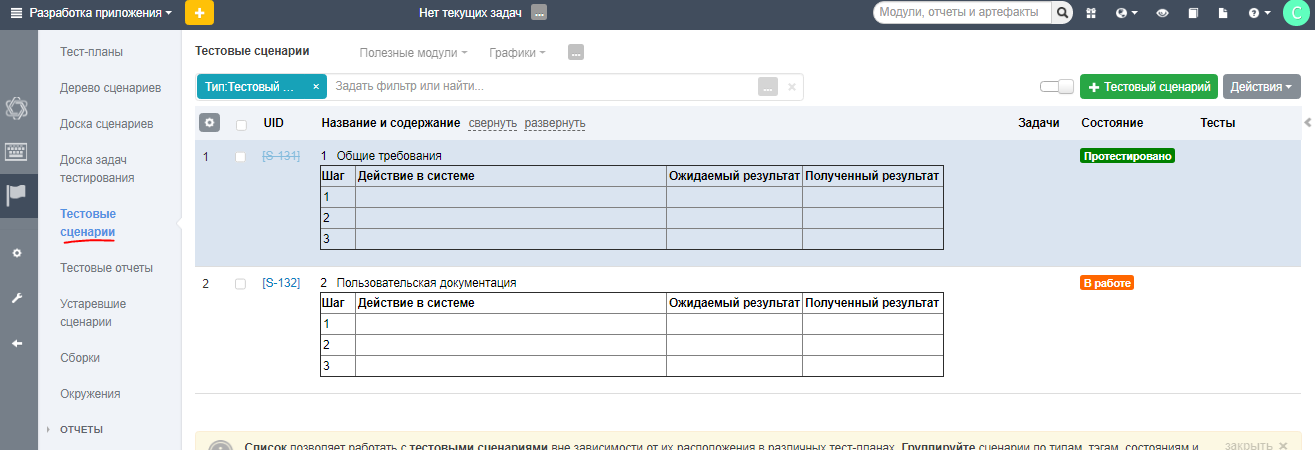


Рисунок 4.23 - Раздел "Тестовые сценарии"

Раздел "Тестовые отчеты" аналогичен по своему назначению разделу из меню "Разработка".

Раздел "Устаревшие сценарии" содержит в себе список сценариев тестирования, которые потеряли свою актуальность в результате доработки и изменений разрабатываемого программного обеспечения.

Разделы "Сборки" и "Окружения" тоже аналогичны по назначению подобным разделам из меню "Разработка".

По результатам знакомства с пробной версией системы Devprom ALM можно сделать следующее заключение: использование такого инструмента в разработке программного обеспечения действительно поможет грамотно расставить приоритеты в работе проекта, каждый участник проекта будет понимать свою роль в проекте и эффективно выполнять возложенную на него задачу. С такой системой всегда будет возможность иметь оперативную информацию обо всех циклах работы проекта, незамедлительно исправлять ошибки до их перерастания в серьезную проблему и разрабатывать качественное программное обеспечение.

Система Devprom ALM предоставит все необходимые возможности в помощь коммуникациям по проекту, важным процессам и данным проекта.

# Безопасность жизнедеятельности

В своей повседневной деятельности человек может подвергаться воздействию различных опасных ситуаций. Такое может происходить и во время трудовой деятельности. Опасные ситуации всегда несут вред здоровью, но могут иногда и привести к смерти человека.

Поэтому, чтобы уберечь свою жизнь и свое здоровье, человек в любой своей деятельности должен соблюдать определенные правила по технике безопасности.

Цель работы над разделом: анализ рабочего помещения отдела Управления информационными технологиями Иркутского авиационного завода, вывод рекомендаций по устранению опасных факторов на производстве, анализ электробезопасности и пожарной безопасности.

Задачи:

1. Дать характеристику объекту;
2. Провести анализ вредных и опасных факторов;
3. Выдать рекомендации для приведения в норму микроклимата в помещении;
4. Провести анализ электробезопасности;
5. Провести анализ пожарной безопасности;

## Характеристика объекта

Помещение отдела Управления информационными технологиями находится в одном из зданий, расположенных на территории Иркутского авиационного. Возле здания проложена асфальтированная дорога для транспорта предприятия. Также имеются подключение к электрическим сетям, сетям водоснабжения, проведены телефонные линии.

Здание может подвергаться воздействию сейсмической активности в результате землетрясений, происходящих на озере Байкал. Вероятность того, что здание будет затоплено небольшая, так как здание располагается на большом удалении от реки.

Предприятие, на территории которого находится здание, оказывает воздействие на окружающую среду, но для снижения вреда от подобных воздействий на предприятии принимаются различные комплексы мер: реконструируются вредные производства, устанавливается современное оборудование, внедряются экологически чистые технологии и т. п.

В помещении располагаются персональные компьютеры и другая оргтехника, которой пользуются инженеры-программисты для выполнения своих трудовых обязанностей.

Технические характеристики ПК: тактовая частота процессора- 3100 МГц, 4 ядра, объем оперативной памяти- 4 Гб, объем жесткого диска- 500 Гб, блок питания имеет мощность 500 Вт. На компьютерах установлена операционная система Windows 7. Помимо системного блока, на рабочем месте есть монитор, устройства ввода (мышь, клавиатура, сканер), а также устройства вывода: принтер. Все устройства подключены к сети 220 В.

## Анализ вредных и опасных факторов

В процессе своей работы человек подвергается влиянию множества факторов, таких, например, как влажность, температура, потоки воздуха и другие.

Данные факторы оказывают влияние на производительность труда и состояние здоровья работника предприятия.

В данном разделе приведены результаты анализа вредных и опасных факторов, которые регламентируются нормативными документами ГОСТ 12.0.003-2015 [20] и Р 2.2.2006-05 [21]. Результаты анализа приведены в таблице 1:

Таблица 1 - Результаты анализа вредных и опасных факторов

|  |  |  |  |
| --- | --- | --- | --- |
| Опасные и вредные  факторы | Источники, места, причины возникновения | Нормируемые  параметры | Основные  средства защиты |
| Вредные факторы | | | |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| Отклонение параметра микроклимата от нормируемых значений | Недостаточный уровень вентиляции | Категория работ Iа  Холодный период: 𝑡=22−24℃  𝜑=40−60%  𝜗=0,1 м/с  Тёплый период: 𝑡=23−25℃  𝜑=40−60%  𝜗=0,1 м/с  СанПиН 2.2.4.548-96 [23] и ГОСТ 12.1.005-88 [22] | Применение кондиционера или вентилятора |
| Отклонение параметров освещенности в рабочем помещении | Небольшие оконные проемы, источники искусственного освещения расположены неправильно/ недостаточное их количество | 𝑒𝑛=1−1,2%,  𝐸𝑛=200−300 лк.  СанПиН 2.2.2/2.4.1340-03 [24] | Установка дополнительных приборов освещения |
| Повышенный уровень шума в помещении | Шум, издаваемый персональным компьютером, шум от клавиатуры и/или компьютерной мыши | Не более 50дБ.  СанПиН 2.2.2/2.4.1340-03 [24]. | Установить в компьютеры более тихие системы охлаждения, использовать более тихие комп. клавиатуры и мыши |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Электромагнитные излучения | Приборы, которые излучают электромагнитные излучения | E=8-15 В/м,  Н=200 А/м на расстоянии 30 см, Н=10 А/м на расстоянии 50 СанПиН 2.2.2/2.4.1340-03 [24]. | Использовать оборудование, от которого исходит малый процент электромагнитного излучения | |
| Опасные факторы | | | | |
| Опасность поражения электрическим током | Повреждение электрооборудования, нарушение целостности проводки | Сила тока не более 0,6-1.5мА при прохождении через тело человека- слабый зуд, пощипывание кожи).  ГОСТ 12.1.009-2017 [26] | | Правильная и безопасная эксплуатация электроприборов, своевременное устранение неисправностей проводки |
| Опасность возникновения пожара | Повреждение оборудования, нарушение целостности проводки, несоблюдение правил пожарной безопасности | Категорирование  производственного  помещения по  пожароопасности  СП 12.13130.2009  [25] | | Пожарная сигнализация, огнетушитель, пожарные датчики |

## Приведение в норму производственного микроклимата

Человек, который работает за компьютером, подвержено влиянию различных неблагоприятных условий: повышенная или пониженная температура воздуха, повышенная или пониженная влажность, повышенное электромагнитное излучение и т.д.

Для определения допустимых и оптимальных значений показателей микроклимата, необходимо определить категорию работ по степени тяжести. Согласно таким документам как ГОСТ 12.1.005-88 [22] и СанПиН 2.2.4.548-96 [23], определим категорию работ инженера - программиста как Iа. Данная категория работ подразумевает, что в основном работник находится в сидячем положении и прилагает незначительные физические усилия в работе.

В таблице 2 и 3 указаны оптимальные и допустимые значения показателей микроклимата соответственно, исходя из задействованных документов и категории работ Iа. В таблице 4отражены параметры фактические.

Таблица 2 - Оптимальные параметры микроклимата

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Период года | Категория работ по уровню энергозатрат | Температура воздуха, ºС | Температура поверх., ºС | Относ. влажность воздуха, % | Скорость движения воздуха,  м/с |
| Холодный | Ia | 22-24 | 21-25 | 40-60 | 0,1 |
| Теплый | Ia | 23-25 | 22-26 | 40-60 | 0,1 |

Таблица 3 - Допустимые параметры микроклимата

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Период года | Категория работ по уровню энергозатрат | Температура воздуха, ºС | | Температура поверх., ºС | Относит. влажность воздуха, % | Скорость движения воздуха,  м/с | |
| ниже | выше | ниже | выше |
| Холодный | Ia | 20-21,9 | 24,1-25 | 19-26 | 15-75 | 0,1 | 0,1 |
| Теплый | Ia | 21-22,9 | 25,1-28 | 20-29 | 15-75 | 0,1 | 0,2 |

Таблица 4 - Фактические параметры микроклимата

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период года | Температура воздуха, ºС | Температура поверх., ºС | Относ. влажность воздуха, % | Скорость движения воздуха,  м/с |
| Холодный | 22 | 21 | 55 | 0,1 |
| Теплый | 26 | 27 | 53 | 0,1 |

После анализа данных из приведенных выше таблиц можно сделать следующие выводы: в холодное время года температура воздуха и температура поверхностей соответствует оптимальным значениям; в теплое время года температура воздуха и температура поверхностей соответствует допустимым значениям. Скорость движения воздуха и относительная влажность в оба периода соответствует оптимальным параметрам микроклимата.

Для достижения оптимальных микроклиматических условий на рабочем месте сотрудника необходимо установить в помещении кондиционер для охлаждения и фильтрации воздуха в теплый период года. Если в холодный период температура воздуха будет некомфортной, то необходимо установить дополнительные обогревательные приборы.

### Расчет комфортности производственной среды

Для расчета комфортности среды используется комплексный показатель комфортности среды, формула определения 1.3.1.1 которого указана ниже.

εK = Qэз - Qтп = Qэз - (Qизл + QK + Qисп) , (1.3.1.1)

где Qэз – энергозатраты организма, Вт;

Qтп – теплопотери организма, Вт;

Длительное воздействие тепла (εK ) приводит к перегреву организма, потеря тепла (εK ) приводит к понижению температуры тела и ощущению холода.

Наиболее оптимальным состоянием, при котором ε0, то есть организм не перегревается и не переохлаждается.

Величина Qэз обычно принимается по нормам.

Qтп– это значения Qизл, Qk, Qисп при известных параметрах поверхности тела человека, определяются параметрами микроклимата и могут быть рассчитаны по формуле:

(1.3.1.2)

где – температура в градусах Кельвина;

𝑇𝑇- принимается равной средневзвешенной температуре тела человека, 31,5 °С;

𝑇П - температуру поверхностей принять равной температуре воздуха в °К.

𝑃𝑇- соответствующее значение парциального давления насыщенных водяных паров при температуре тела человека, *PТ*= 4,61 кПа;

𝑃𝐵- плотность водяных паров при температуре 𝑇𝐵и относительной влажности 𝐶𝑝, определяется по формуле 1.3.1.3

(1.3.1.3)

где *РН* - парциальное давление насыщенных паров воды.

В таблице 5 можно посмотреть результаты расчета комфортности среды.

Таблица 5 - Расчет суммарных теплопотерь организма

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Исходные Данные | | | Параметры микроклимата | | | | | | | | Составляющие теплопотерь | | | |
| Fизл, м2 | FК, м2 | Fисп, м2 | ТП,0С | Кизл | Кисп, Вт/м2 | tВ, 0С | V, м/с | φн, % | РН, кПа | РВ, кПа | Qизл, Вт | Qисп, Вт | Qк, Вт | Qтп, Вт |
| 1,7 | 1,5 | 1,9 | 26 | 4,6 | 4,3 | 26 | 0,1 | 53 | 3,166 | 1,677 | 51,69 | 23,94 | 33,83 | 109,46 |

Расчеты:

𝑄эз = 139 Вт

*==* 51,69 Вт

𝑄𝑘 = (𝐾𝐾 × 𝐹𝐾 × (𝑇𝑇 − 𝑇𝐵)) = 4,1 1,5(31,5 − 26) = 33,83 Вт

𝑄исп = (𝐾исп × 𝐹исп × (𝑃𝑇 − 𝑃𝐵)) = 4,3 1,9(4,61 − 1,68) = 23,94 Вт

𝑄𝑚𝑛 = 𝑄изл + 𝑄𝑘 + 𝑄исп = 51,69+ 33,83+ 23,94 = 109,46 Вт

𝜀 = 𝑄эз − 𝑄𝑚𝑛 = 139 – 109,46 = 29,54 Вт

Комфортность среды 𝜀 = 29,54 Вт. Значение является положительным, поэтому при длительном воздействии тепла может произойти, перегрев организма.

## Электробезопасность

На рабочем месте инженера - программиста находится системный блок, монитор, устройства ввода (клавиатура и мышь), а также другое необходимое оборудование.

Перед началом работы работник должен осмотреть оборудование и убедиться, что оно полностью исправно. При работе за компьютером на работника могут оказывать влияние такие факторы как электромагнитное излучение, статическое электричество, высокое напряжение.

От воздействий сильного электромагнитного излучения поможет приобретение и установка оборудование, у которого уровень такого излучения минимален.

Для предотвращения воздействия статического электричества можно использовать увлажнители воздуха, либо при их отсутствии чаще проводить влажную уборку и проветривание помещения.

От воздействия высокого напряжения необходимо: заземлять электрооборудование, не стоит проникать внутрь электрооборудования во время его работы, чтобы избежать поражения электрическим током, не рекомендуется работать за компьютером с мокрыми руками или в мокрой одежде и пользоваться компьютером, если есть какие - либо неисправности.

Примерная схема заземления компьютера показана на рисунке 5.1.

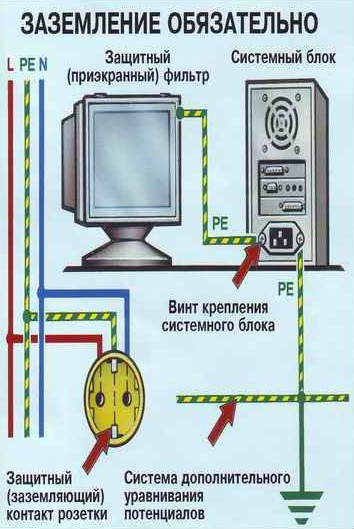


Рисунок 5.1 - Схема заземления

## Пожарная безопасность

Рабочее место с ЭВМ является источником возгорания в случае короткого замыкания. Помимо короткого замыкания возгорание может возникнуть при перегреве элементов компьютера, неисправности проводки и нарушения техники пожарной безопасности. При большом количестве ЭВМ в одном помещении риск возникновения пожара сильно увеличивается. Поэтому необходимо соблюдать меры пожарной безопасности.

Меры пожарной безопасности - это действия по обеспечению пожарной безопасности, в том числе по выполнению требований пожарной безопасности.

Для предприятий организация пожарной безопасности регламентируется следующими документами: федеральные законы № 123-ФЗ[27] и 69-ФЗ[28].

Обеспечение пожарной безопасности является важной частью в охране труда, и оно подразумевает выполнение следующих мероприятий:

* Проведение различных мероприятий по предупреждению пожара;
* Использование средств пожаротушения и оповещения;
* Обучение работников правила пожарной безопасности, проведение инструктажей.

Помещение отдела Управления информационными технологиями находится в двухэтажном железобетонном здании. В самом помещении располагается пару десятков ЭВМ, а также другое необходимое электрооборудование.

Для предотвращения пожара в помещении установлены противопожарные датчики и система противопожарной сигнализации. Для эвакуации персонала в помещении также имеются в наличие эвакуационные выходы. Также в помещении расположены места для углекислотных огнетушителей ОУ - 5.

Принцип применения данного огнетушителя следующий:

1. Сорвать пломбу, выдернуть чеку;
2. Перевести раструб в горизонтальное положение и нажать рычаг;
3. Направить струю заряда на огонь.

### Мероприятия по эвакуации

Эвакуация людей при пожаре – процесс организованного самостоятельного движения людей наружу или в безопасную зону из помещений, в которых есть возможность воздействия на людей опасных факторов пожара [29].

В случае возникновения пожара и срабатывания системы пожарной безопасности, работник должен:

1. Сообщить о пожаре по телефону "01" или "112";

2. Покинуть здание согласно плану эвакуации;

3. По возможности отключить электроснабжение и вентиляция;

4. По возможности приступить к тушению пожара первичными средствами пожаротушения (огнетушители, песок, земля, вода, асбестовое полотно);

5. По прибытию пожарных расчетов встретить их и сообщить о том, где и что горит;

## Выводы по разделу

При выполнении работ по данному разделу были проанализированы условия труда инженера - программиста отдела Управления информационными технологиями на основании факторов производственного микроклимата.

Было дано описание объекта, проведен анализ вредных и опасных факторов. Также были даны рекомендации по приведению в норму микроклимата помещения. Кроме того, были рассмотрены вопросы, связанные с электробезопасностью и пожарной безопасностью.

# Заключение

В результате выполнения работы была описана основная суть системы ALM, ее основные особенности, назначение, достоинства и недостатки применения при разработке программного обеспечения.

Также были описаны существующие решения класса ALM, которые можно применять в разработке программного обеспечения для создания качественного и оптимизированного продукта.

В качестве платформы ALM, которую можно применить для разработки программного обеспечения в отделах Управления информационными технологиями Иркутского авиационного завода, можно предложить программный продукт Devprom ALM.

Тестирование демонстрационной версии данной системы показало, что она может стать хорошим решением для оптимизации, координации и разработки программного обеспечения различного уровня сложности и весьма удовлетворяет современным требованиям в получении качественного и оптимизированного программного продукта.

# Список использованных источников

1. Статья "Обзор управления жизненным циклом приложений с Microsoft Power Platform” [Электронный ресурс] URL: https://docs.micro-

soft.com/ru-ru/power-platform/alm/overview-alm (дата обращения 25.05.20).

1. Статья "Система управления жизненным циклом программного обеспечения" [Электронный ресурс] URL: https://studopedia.net/14\_59454\_sistema-upravleniya-zhiznennim-tsiklom-programmnogo-obespecheniya.html (дата обращения 25.05.20).
2. Статья "Понятие методологии управления жизненным циклом приложений - ALM" [Электронный ресурс] URL: https://megaobuchalka.ru/10/25372.html (дата обращения 25.05.20).
3. Статья "ALM - платформа Borland" [Электронный ресурс] URL: https://www.bytemag.ru/articles/detail.php?ID=6642 (дата обращения 25.05.20).
4. Статья "Программный круговорот" [Электронный ресурс] URL: https://www.osp.ru/os/2011/01/13006960/ (дата обращения 25.05.20).
5. Статья "What is ALM?" [Электронный ресурс] URL: https://download.microsoft.com/download/9/5/F/95FB4775-F769-477E-B187-6737DC09BC9D/What%20is%20ALM--Chappell.pdf (дата обращения 25.05.20).
6. Статья "Различие между ALM и SDLC" [Электронный ресурс] URL: https://www.quora.com/What-is-the-difference-between-ALM-and-SDLC (дата обращения 25.05.20).
7. Статья "Application LifeCycle Management" [Электронный ресурс] URL: http://en.wikipedia.org/wiki/Application\_lifecycle\_management. (дата обращения 25.05.20).
8. Статья "Управление жизненным циклом приложений" [Электронный ресурс] URL: https://ru.qwe.wiki/wiki/Application\_lifecycle\_management (дата обращения 25.05.20).
9. Статья “Different Approaches to an Integrated ALM” [Электронный ресурс] URL: https://ru.calameo.com/books/0004752161a6501a74d2c (дата обращения 25.05.20).
10. Georgi A. Markov, Olga R. Druzhinina. Towards an industrial ALM (Application Lifecycle) Tool Integration, 2011 - 72 с.
11. Презентационный материал "The Changing Face Of Application LifeCycle Management" by Carey Schwaber [Электронный ресурс] URL: https://pdfslide.net/education/serena-life-cyclemanagement.html (дата обращения 25.05.20).
12. Kääriäinen J., Välimäki A. (2008) Impact of Application Lifecycle Management — A Case Study. [Электронный ресурс] URL: https://link.springer.com/chapter/10.1007/978-1-84800-221-0\_5 (дата обращения 25.05.20).
13. Статья "Microsoft Visual Studio Team System" [Электронный ресурс] URL:http://www.tadviser.ru/index.php/Продукт: Microsoft\_Visual\_Studio\_Team\_System (дата обращения 25.05.20).
14. Официальный сайт Devprom ALM [Электронный ресурс] URL: https://www.devprom.ru/company/products/ (дата обращения 25.05.20).
15. Обзор платформы SwiftALM [Электронный ресурс] URL: https://8d9.ru/program/swiftalm (дата обращения 25.05.20).
16. Статья "Открытое управление жизненным циклом приложений (ALM)" [Электронный ресурс] URL: http://www.interface.ru/home.asp?artId=8633(дата обращения 25.05.20).
17. Статья "SAP Solman - Обзор" [Электронный ресурс] URL: https://coderlessons.com/tutorials/sap/izuchite-sap-solman/sap-solman-kratkoe-rukovodstvo (дата обращения 25.05.20).
18. Презентационный материал "Программные средства для управления жизненным циклом приложений" [Электронный ресурс] URL: https://presentacii.ru/presentation/programmnye-sredstva-dlya-upravleniya-zhiznennym-ciklom-prilozhenij-Application-lifecycle-management-ALM (дата обращения 25.05.20)
19. ГОСТ 12.0.003-2015.Система стандартов безопасности труда. Опасные и вредные производственные факторы. Классификация.
20. Р 2.2.2006-05. Руководство, по гигиенической оценке, факторов рабочей среды и трудового процесса. Критерии и классификация условий труда.
21. ГОСТ 12.1.005-88. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
22. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений.
23. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
24. СП 12.13130.2009. Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности.
25. ГОСТ 12.1.009-2017.Система стандартов безопасности труда. Электробезопасность.
26. Федеральный закон "Технический регламент о требованиях пожарной безопасности" от 22.07.2008 № 123-ФЗ.
27. Федеральный закон "О пожарной безопасности" от 21.12.1994 № 69-ФЗ (ред. от 27.12.2019)
28. Официальный сайт компании "КонсультантПлюс" [Электронный ресурс]URL: http://www.consultant.ru/document/cons\_doc\_LAW\_7\_7869

9/faceb5076dc3bfec940ab70185591f70543d6240/ (дата обращения 14.06.20)