Методически рекомендации по разработке второй версии Среды интеграции Экономического факультета МГУ

Оглавление

[Термины и определения 2](#_Toc416272533)

[1. Этапы и модель жизненного цикла 3](#_Toc416272534)

[1.1. Ожидания и требования 4](#_Toc416272535)

[1.2. Архитектура 4](#_Toc416272536)

[1. Адаптеры 5](#_Toc416272537)

[2. Сервер приложений 5](#_Toc416272538)

[3. База данных 6](#_Toc416272539)

[4. Веб сервер 6](#_Toc416272540)

[5. Взаимодействие компонент 6](#_Toc416272541)

[1.3. Техническое задание и технический проект 7](#_Toc416272542)

[1.4. Реализация задачи 7](#_Toc416272543)

[1.5. Приемка и проверка 8](#_Toc416272544)

[1.6. Сопровождение 8](#_Toc416272545)

[2. Управление проектом 8](#_Toc416272546)

[2.1. Аудит проекта 8](#_Toc416272547)

[2.2. Риски 8](#_Toc416272548)

Термины и определения

**Модель последовательного выделения ресурсов –** модель жизненного цикла проекта, при котором целевая система разрабатывается последовательно, совокупность ее функций делится на задачи. Вначале проекта реализуются этап обследования и оформляются требования к системе. На основе требований к системе описывается архитектура. Затем реализуются и сдаются в эксплуатацию отдельные задачи, которые должны удовлетворять требованиям архитектуры.

**Этапы проекта** – проект включает три этапа: Проектирование, Разработка задач, Эксплуатация. На этапе Проектирования реализуется две стадии: формование Требований к системе, разработка Архитектуры. На этапе Разработка задач выполняются этапы технического задания, технический проект, создание решения, интеграция, тестирование, введение в эксплуатацию. Этап эксплуатации включает две стадии - Приемка, Сопровождение.

**Заинтересованная сторона** – это лицо, группа лиц или подразделение организации, имеющая интересы к системе, использованию ее свойств. Интересы Заинтересованной стороны могут быть затронуты как положительно, так и отрицательно в ходе исполнения или в результате завершения проекта.

**Требования** - определение функций системы, отдельных ее компонент, не касаясь внутреннего устройства. Требования к системе формулируется на основе потребностей и ожиданий Заинтересованных сторон. Требования к компонентам определяются на основе трассировки требований к системе на уровень компонент.

**Архитектура** – отражение важных с точки зрения выполнения требований элементов системы, определяющие ее как "прозрачный ящик". Реализуется в виде набора архитектурных решений. Архитектурные решения – решения, которые должны быть устойчивы при расширении функциональности системы и обеспечат ее работоспособность и актуальность в среднесрочной перспективе. В архитектуре должны быть отражены решения накопленные организацией, участниками проекта, аудиторами. В архитектуре могут быть отражены состав компонент, технические решения, организационные практики и т.д. В архитектуре определяются нотации для описания процессов деятельности и модели поставки и обработки данных.

**Техническое задание** – краткое изложение требований, которые должны быть реализованы в рамках определенной задачи. В техническом задании возможно указание модулей архитектуры, задействованных в решении задачи, и перечислены используемые архитектурные решения.

**Задача** - выполнение системой некой функции, в реализации которой участвуют несколько модулей системы. У каждой задачи должен быть функциональный заказчик. Каждая задача реализуется и сдается в эксплуатацию отдельно.

**Технический проект задачи** – описание требований к компонентам, обязательно сопровождаемое схематичных отражением процесса деятельности, процесса сбора и обработки данных и описанием реализуемых сервисов. Процессы деятельности описываются с указанием ролей и выполняемых ими действий. Модель сбора и обработки данных описывает состав сервисов, поставляемые данные и прочие элементы взаимодействия. Вариант решения задачи путем создания новой компоненты модуля или расширение функций созданных компонент выбирается на этапе технического проекта.

**Рабочая группа задачи** – группа людей, созданная для исполнения задачи. В состав рабочей группы входят представитель Заказчика, представитель Исполнителя, руководство проектом, возможно участие представителя Заинтересованной стороны.

**План-график задачи** – утвержденный документ, предназначенный для исполнения и управления реализацией задачи

**Исполнение –** создание новых или модификация существующих программных компонент системы. При создании компонент используются архитектурные решения. При создании решения задачи могут быть использованы созданные ранее компоненты, либо расширен функционал существующих компонент. Созданные Компоненты должны удовлетворять требованиям архитектуры, а также пройти тестирование. При расширении функций модулей необходимо обеспечить работоспособность сданных в эксплуатацию задач.

**Исполнитель** – организация или подразделение, осуществляющее исполнение проекта.

**Интеграция -** разворачивание компоненты для взаимодействия с другими компонентами.

**Проверка решения** - соответствие разработки Требованиям.

**Приемка решения** – соответствие разработки Ожиданиям заказчика.

**Сопровождение** – обеспечение работоспособности и актуализация информационного наполнения системы в процессе ее эксплуатации.

# Этапы и модель жизненного цикла

Для реализации проекта предлагается использовать модель **последовательного выделения ресурсов под каждую задачу.** Вначале выполняются этапы обследования и фиксация ожиданий заказчика, затем они оформляются в форме требований к системе. На основе требований вся предполагаемая система описывается на уровне архитектуры. В архитектуре отражается состав компонент системы, принципы их взаимодействия, определяются требования к архитектурным артефактам, которые обеспечат исполнение требований к системе со стороны Заказчика. Затем разрабатываются отдельные задачи (рис. 1)

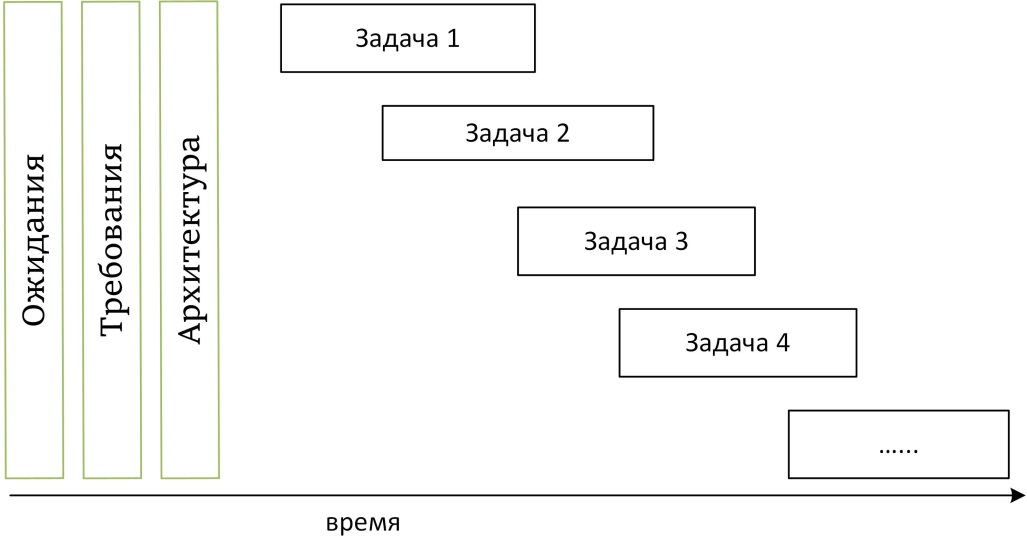


Рисунок Этапы исполнения проекта

Разработка задачи включает формирование рабочей группы, в которую входят представитель функционального заказчика и участники проекта. Рабочая группа готовит Техническое задание, в котором прописывается состав компонент задачи. Затем определяется исполнитель каждой компоненты. Рабочая группа совместно с исполнителем готовит техническую документацию для разработки компонент.

Затем подготавливаются компоненты, тестируются, производится их интеграция. После проверки компонент задача сдается в эксплуатацию.

Состав задач, последовательность их выполнения определяется исходя из текущих потребностей и возможностей. Такой подход к разработке системы позволит сделать ее достаточно динамичной. Архитектура должна включать набор архитектурных артефактов, которые позволят динамично расширять состав задач под изменения в предметной области.

## Ожидания и требования

Основой для формирования архитектуры системы являются ожидания заинтересованных сторон. Поэтому разработку системы предлагается начинать со сбора и анализа информации, отражающей ожидания заинтересованных сторон.

В качестве источников для формирования могут быть следующие документы:

* Программа развития факультета
* Программа развития коммуникационной среды
* Результаты мозгового штурма по развития магистратуры

В дополнения к имеющимся материалам предлагается организовать сбор предложений от административных подразделений факультета, сотрудников, студентов, выпускников в форме предложений. Сбор предложений можно организовать в виде специального ресурса на сайте факультета.

Поступившие со всех источников предложения обрабатываются на основе «Методики формирования портфеля приложений» (Приложение 1). Для ее реализации нужно разработать методику заполнения позиций Ценность, Потенциал, Информация по подразделениям факультета.

## Архитектура

В архитектуре отражаются принципы построения всех основных модулей системы: адаптеров, базы данных, сервера приложений, веб сервера и мобильного приложения, приводится описание архитектурных артефактов: объектная модель, реализация логики предметной области в каждой из компонент, а также базовые принципы их взаимодействия.

Во второй версии Среды интеграции предполагается следующий состав модулей (Рисунок 2): Адаптеры, Север приложений (СП), БД интегрированных данных (БД ИД), Личный кабинет (ЛК), Мобильное приложение (МП) и Веб сайт.

* 1. 

Рисунок Состав модлей

* 1. Изменения по составу модулей по сравнению с перовой версией: предполагается добавить два канала доставки информации пользователям: мобильное приложение и веб сайт. Мобильное приложение является модулем полностью разрабатываемым в проекте. Веб сайт является внешней по отношению к создаваемой системе приложением, с возможностью тесной интеграции, в том числе с перераспределением логики действий между СП и ВС.
  2. Модули предполагается реализовать в форме набора самостоятельных компонент, которые создаются для решения отдельных задач и взаимодействуют с другими компонентами через программный интерфейс (API).
  3. На этапе Архитектуры определяется набор возможных технологий для реализации каждого модуля системы, а также платформы разработки, используемые языки программирования, и прочие важные артефакты.

### Адаптеры

Адаптеры создаются для взаимодействия с внешними системами. Работой адаптера управляет администратор. Администратору системы должна быть предоставлена возможность определять параметры работы адаптера, а также определять его характеристики в зависимости от типа вызова: по расписанию, по требованию, по событию.

### Сервер приложений

Основным исполнителем бизнес логики является сервер приложений. Существенной частью сервера приложений является объектная модель. Объектная модель описывает структуру объектов, их атрибуты, методы, связи с другими объектами. В объектной модели детально отражаются понятия и объекты реального мира, представленные в концептуальной модели. Объекты используются для кода: посредством методов объектов и программного кода реализуется большая часть бизнес логики.

Логика деятельности реализуется в основном путем предоставления работы с веб сервисами отдельным ролям.

При реализации второй версии необходимо разработать инструмент мониторинга работоспособности всех компонент и сервисов СП Администратором системы.

### База данных

Основной частью базы данных является модель данных, которая отражает организацию хранения данных в базе. На уровне модели данных закладываются возможности системы по работе с данными, прагматика системы. Необходимо чтобы совокупность данных, хранящихся в системе, позволяла полностью реализовать все требования разрабатываемой системы.

Модель данных Ядра системы строится по принципу Local as View, который предполагает представление модели в терминах заданного интегрирующего глобального представления. Данные каждого из локальных источников данных определяется в терминах общей модели. Это позволит обеспечить динамичность состава множества источников данных.

При таком подходе одним из компонентов сервера приложений должен выполнять верификацию поставляемых из внешних систем данных на соответствие требованиям объектной модели.

Разработанная в первой версии модель данных Ядра системы может быть использована в следующей версии и расширена с учетом требований к системе.

При расширении состава внешних систем и появлением новых задач необходимо будет производить модификацию модели данных. Модель данных Ядра системы должна предполагать возможность реализации большинства задач для удовлетворения требований к системе, а также предоставлять возможность по расширению состава данных.

### Веб сервер

При разработке веб сервера необходимо предусматривать описание архитектры веб сервера. Новая платформа для создания Среды интеграции предполагает более тесное взаимодействие Сервера приложении и Веб сервера.

Дизайн пользовательского интерфейса должен соответствовать требованиям пользователей и использовать современные технологии. Для функций формирования HTML-кода использовать стандартные библиотеки и общепризнанные платформы разработки.

### Взаимодействие компонент

Предоставление сервисов внешним пользователям предполагает последовательность взаимодействий между модулями и компонентами системы. Порядок взаимодействия должен быть регламентирован. Регламентация реализуется путем составления контракта.

**Контракт** это набор правил, которые должны выполнять все участники контракта. Участников контракта обычно двое или трое, в первом случае это поставщик и получатель, во втором – поставщик, получатель и заказчик (получатель комплексной услуги или продукта).

Условия контракта должны быть закреплены всеми сторонами. В случае необходимости в контракт могут быть внесены изменения.

## Техническое задание и технический проект

После утверждения архитектуры определяется последовательность выполнения задач проекта. Состав функций задачи, сроки, финансирование определяется на этапе принятия решений о его разработке, после этого идут этапы Технического задания и Технического проекта, разработки компонент, их интеграции и ввода в эксплуатацию.

На стадии инициирования задачи и составления технического задания определяется состав рабочей группы. В техническом задании указываются решаемые задачи, поставщики информации, потребители информации, доступная информация, а также возможно указание архитектурных артефактов, задействованных в решении задачи и перечисление используемых архитектурных решений.

После составления ТЗ определяется исполнитель по каждому модулю или компоненте задачи.

Далее составляется **технический проект** задачи, в котором детально описываются требований к компонентам. Технический проект снабжается моделями процесса деятельности и процесса сбора и обработки данных. В моделях указываются роли и выполняемые ими действия.

В составлении технического проекта может быть задействован представитель исполнителя.

В состав Технического проекта входят Контракты по взаимодействию и План-график выполнения задачи.

## Реализация задачи

Реализация задач предполагает использование технологий, определенных на стадии архитектуры. В состав поставляемого решения входит источники кода, а также состав компонент. Код и компоненты должны быть снабжены документацией и описанием.

**Ориентировочный список задач на основе первой версии**

«Актуальный список сотрудников»

«Актуальный список студентов»

«Учебные планы»

«Аккредитация студентов»

«Аккредитация преподавателей»

«Ведомости и рейтинг»

«Научное руководство и Антиплагиат»

«Статистика и показатели»

и т.д.

## Приемка и проверка

**Приемка и проверка** выполняется как для системы в целом, так и для отдельных модулей, компонент, ролей, сервисов и методов. При проверке и приемке системы в качестве заказчика выступает Заинтересованная сторона, в случае создания отдельных архитектурных компонент, методов и сервисов системы заказчиком выступает получатель сервиса.

Сдача в эксплуатацию предполагает обязательное наличие Руководства пользователя.

## Сопровождение

После приемки задачи в эксплуатацию осуществляется сопровождение системы, осуществляемое подразделением факультета

# Управление проектом

Для управления проектом используется инструмент, позволяющий определять задачи и сроки и отслеживать их выполнение, т.е. выполнять план- факт анализ

Варианты приложения:

* Sharepoint
* MS Project
* Собственное решение
* <https://www.wrike.com/>

## Аудит проекта

Аудит выполняют внешние эксперты. Необходимо провести аудит Первой версии, методических разработок для второй версии, а также после выполнения соответствующих стадий аудит Требований к системе, аудит Архитектуры и далее выборочный аудит задач, включая все этапы: ТЗ, техпроект, исполнение, проверка приемка, сдача в эксплуатацию.

## Риски

* **Законодательные** – персональные данные, использование нелицензионного ПО разработки
  + Необходимо упорядочить и привести в соответствие текущему законодательству работу с персональными данными
* **Бюджетные** – реализация проекта в полной мере требует поиск дополнительного финансирования. Метод управления – качественные документы по требованиям и архитектуре для привлечения внешнего финансирования
* **Организационные** – смена Исполнителей отдельных модулей и компонент, поэтому нужно определить технологии и платформы разработки, которые не привязаны к исполнителям компонент. Метод управления - Важным является этап принятия Архитектуры. Архитектура должна быть основным элементом контроля проекта. Архитектура разрабатывается на основе требований.
* **Архитектурные** – неправильный выбор технологий реализации, ошибки концептуальной модели. Метод управления - необходим постоянный внешний аудит.
* **Технологические** –информационная безопасность (нужно прорабатывать).
* **Квалификационные** – навыки по использованию современных технологий участниками проекта. Метод управления - внешний аудит технологий.