

# Заявка в номинацию весеннего конкурса образовательных инноваций 2023 года

Номинация: Практика внедрения командной/проектной работы студентов магистратуре

Преподаватели:

Песоцкая Елена Юрьевна, Хади Салех Мухаммед

Название: Проектная работа студентов в магистратуре (1 год обучения)

**Аннотация.** Дисциплина «Методология программной инженерии» / Software Engineering Methodology (читается на англ. яз) направлена на *развитие у студентов теоретических и практических навыков, знакомство с реальными технологиями и инструментами разработки, взаимодействие в команде, работа на протяжении жизненного цикла разработки программного продукта* на первом курсе магистерской программы «Системная и программная инженерия» факультета компьютерных наук. Студенты получают набор теоретических знаний в соответствии с ведущей методологией IEEE, и имеют возможность выполнить практический этап – программную реализацию жизнеспособного продукта.

**Основная цель практики** студенческой команды — совместно разработать программный продукт или сервис, который будет обладать минимальной жизнеспособностью (MVP) и иметь перспективы дальнейшего развития.

## Содержание

Предпосылки внедрения проектной деятельности и план организации занятий.....	2
Способы взаимодействия со студентами.....	4
Организация обратной связи от студентов .....	6
Порядок оценивания работ .....	6
Способы масштабирования представленной технологии оценивания с помощью цифровых технологий.....	9
Приложение 1. Пример итоговых результатов (презентация решения).....	10
Приложение 2. Пример документа SRS (требования).....	11
Приложение 3.Демо приложения.....	11

## Предпосылки внедрения проектной деятельности и план организации занятий

Дисциплина «Методология программной инженерии» входит в блок обязательных дисциплин на первом курсе и длится два модуля — с сентября по декабрь. В рамках данной дисциплины студенты магистратуры получают необходимую базу в части программной инженерии на основании методологии и стандарта IEEE, знакомятся со всеми процессами разработки ПО на протяжении всего жизненного цикла.

Дисциплина не является новой, читается студентам на протяжении последних нескольких лет, регулярно совершенствуется и дорабатывается в части обновления стандартов и методологии.

В то же время, было выявлено, что ценность дисциплины и изучения методологии достаточно ограничена в виду теоретизированности знаний и отсутствия возможности практики у студентов.

В качестве внедрения практикоориентированного подхода в 2020 году студентам был предложен подход, в рамках которого студенты получили возможность объединяться в команды и заниматься совместной разработкой программного продукта/приложения/сервиса.

Результатом такой работы является полностью выполненный ИТ-проект, с необходимым комплектом в соответствии со стандартами IEEE. В 2021 году практический этап был доработан на основе наблюдений и обратной связи от студентов по результатам прошлого года, а также с учетом факта инфляции оценок.

Практическая работа над проектом ведется в течение двух месяцев и разбита на несколько этапов в соответствии с теоретическим материалом, предложенным на лекциях.

В таблице 1 представлен **План организации занятий** и перечислены ключевые этапы работы над проектом и их результаты.

#	Теоретический блок	Практический этап	Образ результата	Метод оценивания
1	Введение в программную инженерию. Цели, задачи, требования к специалистам.	<b>Объединение в команды</b> 3-4 чел и выбор проекта для реализации.	Project Brief. Бриф с идеей проекта. Распределение ролей в команде, высокоуровневые задачи команды.	Защита идеи перед другими студентами и преподавателями.
2	Сбор и формализация требований к ПО, анализ целей заказчика.	<b>Работа с требованиями.</b> Анализ целевой аудитории и потребностей пользователей. Изучение лучших практик. Создание списка функциональности и приоритезация. Ограничения решения.	Software Requirements Specification. Документ с формализацией функциональных требований (ТЗ) и нефункциональных требований	Защита качества документации и представление ключевой функциональности по проекту.
3	Управление разработкой. Выбор	<b>Разработка ПО.</b> Выбор средств поддержки	Task Manager. Демонстрация плана	Защита выбора средств разработки

	средств разработки, организация процесса, работа в команде.	процесса разработки (Trello/Jira, пр), создание плана разработки, выбор инструментов и средств реализации.	задач в выбранной системе, обоснование методов и средств разработки.	и выбора системы поддержки разработки ПО.
4	Архитектура и дизайн, тестирование продукта	<b>Планирование архитектуры и тестирование</b>	Architectural Design. Схематичное представление процессов, высокоуровневая схема архитектуры решения. Тестирование ПО	Защита архитектуры решения, представление результатов тестирования
5	Внедрение ПО, эволюция и сопровождение.	Подготовка прототипа ПО, <b>защита решения</b>	Готовый прототип	Демонстрация готового решения – жизнеспособного ПО (MVP)

*Табл. 1. План организации занятий*

Далее рассмотрим более детально практические задачи каждого этапа.

#### *Этап 1. Распределение проектов, организация команд.*

Студентам предлагается самостоятельно организовать команды 3-4 человека и выбрать проект для реализации.

Студентам доступна таблица для записи, где они могут вписать своё имя и дать информацию об имеющихся навыках и опыте работы. Например BackEnd programmer, ML, data analyst.

В качестве инициатив доступен пул проектов, предложенных преподавателями дисциплины (это различные пользовательские сервисы, приложения, платформы). При этом поощряются инициативные темы студентов, которые им интересно выполнять. Преподаватели дают рекомендации по сложности и трудоёмкости предложенного проекта относительно численности команды.

#### *Этап 2. Работа с требованиями*

После того как проекты распределены, студенты приступают к извлечению и формализации требований. К моменту защиты студенты провели полноценный анализ не только аналогов, но и последних технологий, методов и подходов к разработке в этой области/индустрии. Далее разрабатывается техническое задание на реализацию в формате Software Requirements Specification (стандарт IEEE), где описываются в формальном виде ключевые требования.

#### *Этап 3. Управление процессом разработки*

Следующий этап — управление процессом разработки. Команда при желании может использовать определенную Agile-методологию разработки, например Scrum. Обязательным является выбор средств поддержки процесса разработки (Trello/Jira, пр). Желательным условием в ходе реализации является использование системы контроля

версий, таких, например, как GIT/SVN/ Mercurial, что позволяет не только повысить качество реализуемого ПО, но и отследить вклад каждого участника команды.

#### *Этап 4. Проектирование архитектуры и дизайн, тестирование*

На этом этапе студенты определяют жизненный цикл разработки продукта, тип архитектуры, используемые патеры проектирования, базовую структуру программного продукта, выбирают средства и инструменты разработки.

Студенты не ограничены выбором конкретных средств разработки и обуславливают свой выбор с учетом имеющегося опыта и знаний, полученных в предыдущие годы обучения. Однако команды должны обосновать выбор средств разработки и аргументировать применимость этих средств и их эффективность.

В зависимости от специфики своего продукта студенты предлагают план тестирования своего прототипа. Цели тестирования:

1) Выявление багов и составление баг-репорта (можно в свободной форме)

По баг-репортам команда должна сделать небольшой отчет по их исправлению или обоснование почему исправить нельзя.

2) Приёмочное тестирование / тестирование с конечными пользователями на соответствие продукта функциональным требованиям. Проверяем то, что написано в ТЗ на соответствие тому, что сделали.

#### *Этап 5. Представление проекта и защита решения*

Защита происходит в устной форме в виде презентации разработанного проекта с демонстрацией всех этапов управления проектом, проектирования и разработки ПО (слайды), а также с демонстрацией разработанного функционала (представление на компьютере работающего ПО). Также проверяется наличие соответствующей документации. Порядок оценивания представлен далее в разделе «Порядок оценивания работ».

### Способы взаимодействия со студентами

В рамках дисциплины предусмотрены лекции и практические занятия.

На лекциях студенты получают теоретический материал по методологии программной инженерии.

Практические занятия организованы в формате представления результатов командами студентов, обсуждения рекомендаций преподавателя.

Студенты представляют текущий статус и приходят с демонстрацией промежуточных результатов. Также предполагается, что студенты готовят все вопросы, которые их беспокоят в рамках дальнейшей работы.

Помимо регулярных практических занятий предусмотрены консультации по индивидуальным вопросам команд. Команды также могут направлять вопросы преподавателю по электронной почте или мессенджерам.

Общение со студентами организовано через систему SMART LMS, где студенты регулярно выкладывают последние обновления.



### Общение со студентами в Telegram

Там же они получают все материалы, которые им необходимы для работы – образцы брифа проекта, пример SRS документации для заполнения, шаблоны архитектуры или плана тестирования, см. рис. 1.



Рис. 1. Организация материалов в Smart LMS

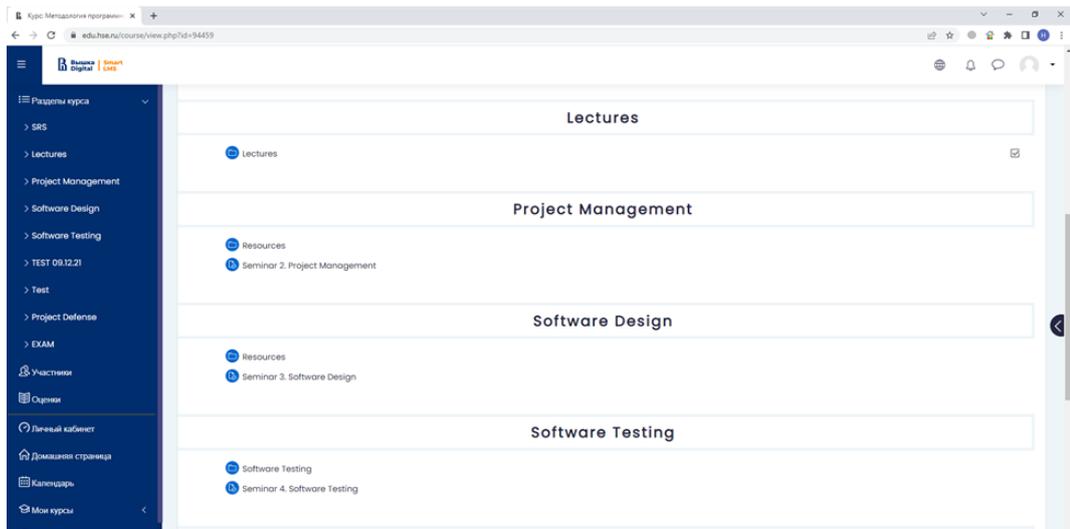


Рис. 2. Организация материалов в Smart LMS

## Организация обратной связи от студентов

Общение со студентами организовано через систему SMART LMS.

В целом студенты отмечали практическую полезность и значимость результатов. Некоторые студенты продолжили выполнения проекта в рамках КР/ВКР. Также ряд студентов отметили, что выполнение подобного рода проектов полезно для будущего опыта и трудоустройства, в качестве примера самостоятельно реализованного полноценного проекта.

Поскольку некоторые студенты уже были трудоустроены на момент обучения на 1м курсе магистратуры, они оценили возможность взять корпоративный проект в качестве инициативы, подключить к проекту своих сокурсников для более качественной и глубокой проработки результатов.

## Порядок оценивания работ

Порядок оценивания состоит из нескольких элементов. Первый элемент текущего контроля и оценивания представляет собой оценку работы по каждому этапу, на котором происходит приемка проделанной работы за каждый этап. Студенты демонстрируют документы в письменной форме по основным этапам работы над проектом и получают оценку по шкале от 1 до 10 за каждый этап. Три этапа промежуточной оценки: SRS (спецификация программных требований), управление процессом разработки и Архитектура + тестирование:

Team	A	B	C	D	E	F	G	H
	Presentation	SRS, требование	Project management	Development+ Architecture +Testin	DEMO + Prototype	Answers to Questions	final	
Organization of Exhibitions	10	10	10	10	10	10	60	
YouPick	7	8	7	8	6	8	44	
SmartFridge	7	8	8	8	8	8	47	
Corona Travel App	8	8	8	7	6	7	44	
Compact Cashbox	7	7	8	8	7	7	44	
NextDoor Farm	7	8	7	7	7	8	44	
Monitoring System for Online Classes	8	8	8	9	8	7	48	
	7	7	8	8	8	7	45	

Итоговый контроль включает также защиту проекта в конце второго модуля и экзаменационный тест.

Формула оценивания:

Итоговая оценка: Активность(А) \* 0.2 + Тест(Т) \* 0.1 + Проект(П) \* 0.5 + Экзамен(Э) \* 0.2

### **I. Активность(А)**

Студент считался активным, если демонстрировал одну из следующих компетенций во время занятия:

- активное слушание и уточняющие вопросы (строго по существу);
- дополнение дискуссии в классе своими идеями или примерами;
- помощь сокурсникам - объяснение сложных моментов с использованием примеров;
- поэтапная сдача материалов:



*Рис. 3. Работа студентов в классе*

### **II. Тест (Т):**

Для студентов существует промежуточный контроль в виде промежуточного теста для оценки теоретической подготовки студентов.

Примеры вопросов теста:

1. Назовите пять моделей, используемых в SDLC.
2. На каком этапе жизненного цикла SDCL стоимость исправления ошибок минимальна?
3. Что такое верификация и валидация?
4. Пример нефункциональных требований?

### **III. Проект(П)**

В конце курса проводится публичная защита студенческих проектов, которые они выполняют в командах. Заранее студенты загружают в SmartLMS техническую документацию, презентацию и матрицу ответственности (RACI matrix).

Защита проекта происходит в устной форме в виде презентации разработанного проекта с демонстрацией всех этапов управления проектом, проектирования и разработки ПО (слайды), а также с демонстрацией разработанного функционала.

На защите преподаватели оценивают:

- презентацию выполненного проекта, включая демонстрацию программного продукта;
- объем выполненных работ;
- ответы на вопросы.

Для оценки вклада каждого участника проекта используется формула:

**Проект:** *Оценка презентации (0,2) + Оценка выполненных работ в соответствии RACI matrix (0,5) + Ответы на вопросы комиссии (0,3)*

**А) Оценка презентации:** каждый участник проекта принимает участие в защите проекта и демонстрирует публично свои результаты.

Структура презентации, ожидаемой от студентов:

- цель и задачи проекта;
- описание проблемы и предметной области проекта;
- функциональные требования и нефункциональные требования;
- бизнес-требования и матрицы ответственности участников;
- план управление проектом и задачи команды;
- проектирование системы и архитектура системы;
- тестирование и верификация решения;
- демонстрация прототипа и результатов.

**Б) Оценка выполненных работ в соответствии RACI matrix:**

В процессе утверждения темы проекта студенты разрабатывают матрицу ответственности (RACI matrix), где указывается зона ответственности каждого участника и его задачи в рамках данного проекта, данная матрица неотъемлемая часть документации проекта.

	Teona Sadulaeva	Vladislav Shchukin	Lyudmila Rezunik
Project description and non-functional requirements	R	C	A
UI Requirements	A	R	I
Upload v.1.0 of SRS	R	I	A
UI prototype	R	I	C
Project Plan	C	I	R
SRS 2.0	A	C	R
API Specification	C	A	R
Technical proposal + RFP	A	R	C
Implement the frontend	R	I	C
Implement basic API calls	C	I	R
Acceptance Test Plan	A	R	C
Set up testing environment (deploy)	I	R	A
Testing (security and stress)	I	R	C



Рис. 4. Матрица ответственности (RACI matrix)

**С) Ответы на вопросы комиссии на защите проекта:** студент отвечает на вопросы, заданные комиссией по своей части выполненных задач данного проекта.

Примеры вопросов, задаваемых комиссией:

1. Какой подход вы использовали при формировании требований.
2. Какие из требований нефункциональные? Как вы собираетесь выполнять проверку этих требований? Какие показатели измерения применимы к нефункциональным требованиям?
3. Почему выбрана именно эта архитектура?
4. Какие задачи выполняются на этапе кодирования?
5. Какой подход к разработке использовала ваша команда? Как планировались итерации/спринты?
6. Какова ЦА вашей разработки? Как вы это учитывали в программном продукте/интерфейсе?

#### IV. Экзамен(Э)

Экзамен проводится в формате оффлайн, на экзамене студенты выполняют тест в системе SmartLMS.

Примеры вопросов экзамена:

1. Назовите этап, на котором отслеживалась производительность новой системы?
2. Назовите два инструмента, которые используются для отслеживания требований к программному обеспечению?
3. Что подразумевается под объемом программного обеспечения?
4. Какие существуют методы оценки программного проекта?

Способы масштабирования представленной технологии оценивания с помощью цифровых технологий

В будущем возможно приглашать специалистов и экспертов для получения независимой оценки, а также отправить работу на рецензию. Работа может быть оценена коллективом департамента программной инженерии. Все предложенные варианты можно реализовать режиме онлайн, либо через платформу Smart LMS

Приложение 1. Пример итоговых результатов (презентация решения)

## Event Management Application

1 **Event Management Application EveLife**

2 **Team**

3 **Problem statement**

4 **Project goal**

5 **Project Description**

6 **Functional requirements**

7 **Kanban board**

8 **Activity timeline**

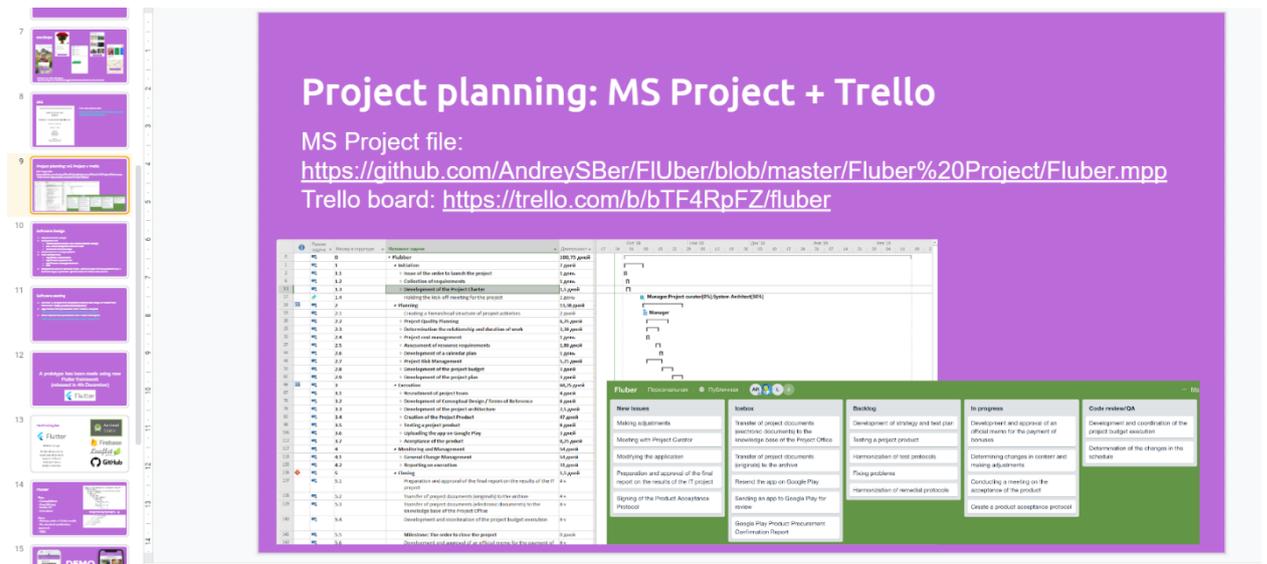
9 **Class diagram**

10 **Software technology**

11 **System design**

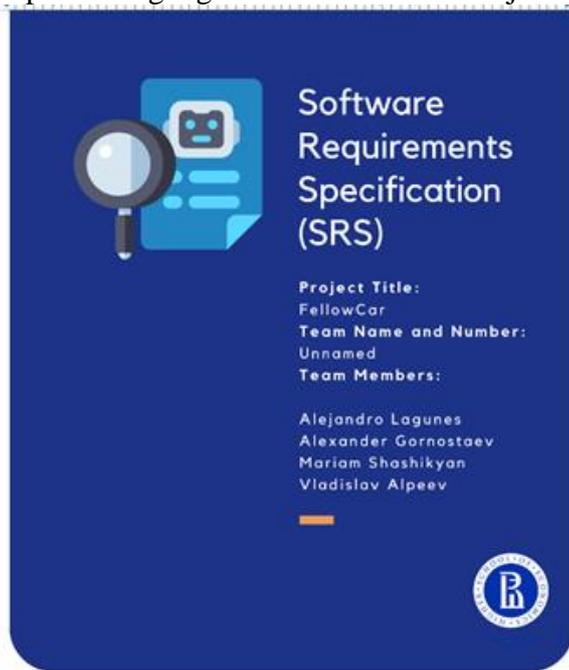
12 **Prototype**

[https://docs.google.com/presentation/d/1E-4rQPKIqfbRII9rQximwllEohYBV\\_2k/edit#](https://docs.google.com/presentation/d/1E-4rQPKIqfbRII9rQximwllEohYBV_2k/edit#)



## Приложение 2. Пример документа SRS (требования)

<https://docs.google.com/document/d/1Aj4nc8vQydSpy-Hvqu7FUUh8RkG1hZVW/edit>



## Приложение 3. Демо приложения

[https://docs.google.com/presentation/d/1Mtg\\_EtFbobj0Ms6Sd2HNLCTCUCeEpeAFqw3E\\_N98F8sjk/edit#slide=id.g295ebf85979\\_0\\_1](https://docs.google.com/presentation/d/1Mtg_EtFbobj0Ms6Sd2HNLCTCUCeEpeAFqw3E_N98F8sjk/edit#slide=id.g295ebf85979_0_1)