

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**КАФЕДРА МО ЭВМ**

**ОТЧЕТ**  
**по практике в компании ЕРАМ**

Студент гр. 8304

\_\_\_\_\_

Рыжиков А.В.

Руководитель

\_\_\_\_\_

Шабунина Е.И.

Санкт-Петербург

2021

## ЗАДАНИЕ НА ПРАКТИКУ

Студент Рыжиков А.В.

Группа 8304

Тема практики: Agile, Git, бизнес-аналитика, тестирование ПО, Android, .NET/C#

Задание на практику:

Изучить Agile, Git, бизнес-аналитику, тестирование ПО, Android, .NET/C#

Сроки прохождения практики: 15.02.2021 – 11.06.2021

Студент

\_\_\_\_\_

Рыжиков А.В.

Руководитель

\_\_\_\_\_

Шабунина Е.И.

## СОДЕРЖАНИЕ

	Введение	4
1.	Agile методология	5
1.1	Agile	5
1.2	Scrum	7
1.3	Kanban	8
2.	Git	10
3.	Бизнес-анализ	11
4.	Автоматическое тестирование	12
5.	Android	14
6.	.NET/C#	18
	Заключение	19

## **ВВЕДЕНИЕ**

Распределенная практика прошла в формате лекций и тренингов по шести модулям. Каждый модуль состоял из 2-5 занятий с куратором – сотрудником ЕРАМ соответствующего направления.

Первые два модуля посвящены работе по agile методологии и с Git. Также удалось познакомиться с профессиями тестировщика и бизнес-аналитика в реальных условиях IT компании. Завершили практику занятия с android и .NET/C# программистами.

## 1. Agile методология

Преподаватели: Olga Postnikova

Занятий проведено: 4

В отличие классического Project Management (PM), когда проект жестко регламентирован заранее установленными требованиями (контрактами), Agile предполагает быстроту реагирования, а также гибкую адаптацию к внешним и внутренним изменениям. Это достигается с помощью итеративной разработки продукта и эффективного межличностного общения. В водопадной (каскадной) модели PM, которая считалась стандартом де-факто, проект состоит из функциональных задач, где каждая последующая работа четко регламентирована и начинается строго после окончания предыдущей, например, тестирование начнется только после того, как написан весь код. Жесткая определенность и обилие регламентирующей документации обуславливают длину производственного цикла. При этом продукт считается готовым лишь после выполнения всех этапов

### 1.1. Agile

**4 ключевые идеи Agile** сфокусированы на гибкости и адаптивности этого подхода:

- **Эффективное взаимодействие** между людьми – базовое средство достижения целей;
- **Реально работающий продукт** является главной ценностью;
- **Изменения**, которые могут повысить качество и конкурентоспособность продукта, **приветствуются** на любом этапе разработки;
- Контрактная, техническая и прочая регламентирующая **документация вторична по значимости** относительно работающего продукта и сотрудничества между участниками проекта.



Эти идеи раскрываются в **12 принципах Agile Manifesto**:

1. работающий конкурентоспособный продукт, удовлетворяющий заказчика — лучший показатель прогресса и измеритель эффективности;
2. оперативная и бесперебойная поставка продукта, удовлетворяющего заказчика;
3. адаптивность продукта к новым требованиям, которые могут повысить его ценность и конкурентоспособность (возможность внесения изменений на любом этапе разработки);
4. простота и прозрачность технических решений, документации, процессов и инструментов, чтобы не создавать лишней работы;
5. частая поставка функционирующего продукта (раз в месяц/неделю или ещё чаще);
6. постоянный темп работы всех участников проекта на протяжении всего его срока;
7. минимизация организационных и информационных барьеров, лучший путь передачи информации — это личный разговор лицом к лицу;
8. тесное и ежедневное общение исполнителей с заказчиком в течении всего проекта;
9. мотивация участников проекта и обеспечение их всеми необходимыми условиями работы, поддержкой и доверием;
10. самоорганизация и самоконтроль команды проекта;
11. непрерывное улучшение профессиональных компетенций команды проекта;
12. систематический анализ и постоянный поиск возможностей оптимизации командной и индивидуальной работы

Главные достоинства эджайл – быстрота, адаптивность и фокус на главном. Отсутствие бюрократии и периодичность поставок работающего продукта с постепенным наращиванием его функциональных возможностей существенно сокращают сроки получения итогового результата. Это особенно важно для бизнеса, т.к. благодаря стремительному выходу на рынок можно быстро занять свободную нишу.

Недостатки Agile являются прямым следствием его достоинств:

- снижение важности регламентирующей и технической документации может привести к ее нерелевантности или даже к фактическому отсутствию;
- краткосрочное планирование не всегда учитывает необходимость масштабирования продукта, что влечет ошибки в архитектуре;
- появление новых требований после нескольких итераций приводит к кардинальным изменениям архитектуры и переделкам уже созданных решений;
- накопление дефектов и снижение качества продуктов вследствие решения проблем самым простым и быстрым, но не всегда самым правильным способом.

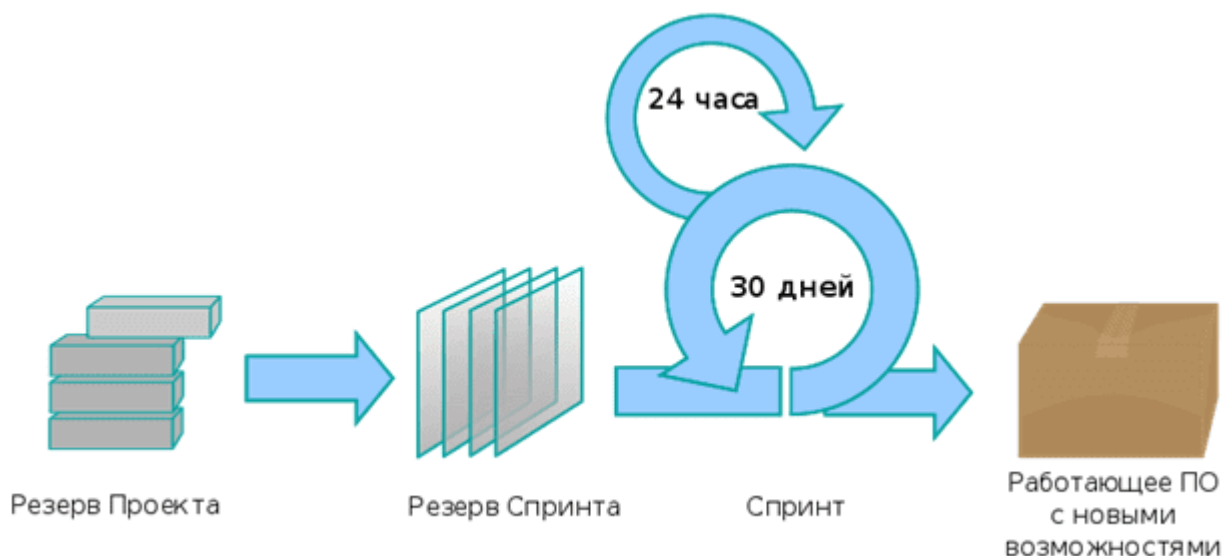


Наиболее популярными Agile-подходами считаются **Scrum** (скрам) и **Kanban** (канбан).

## 1.2. Scrum

В **Scrum** над проектом работает команда профильных технических специалистов (например, аналитик, программист, тестировщик, администратор) вместе с владельцем продукта (product owner) и модератором (scrum-мастер). Product owner аккумулирует бизнес-требования, соединяет команду исполнителей с заказчиком и следит за развитием проекта. Scrum-мастер управляет процессом организации разработки по Agile-принципам: проводит общие собрания (meetings, митинги), мотивирует и поддерживает команду.

В Scrum рабочий процесс делится на равные периоды от 1 до 4-х недель (спринты), в зависимости от проекта и команды. Перед стартом каждого спринта на митинге формулируются его задачи, а в конце обсуждаются результаты. Краткосрочность и измеряемость спринтов позволяет эффективно управлять проектной деятельностью, не перегружая участников проекта авралами

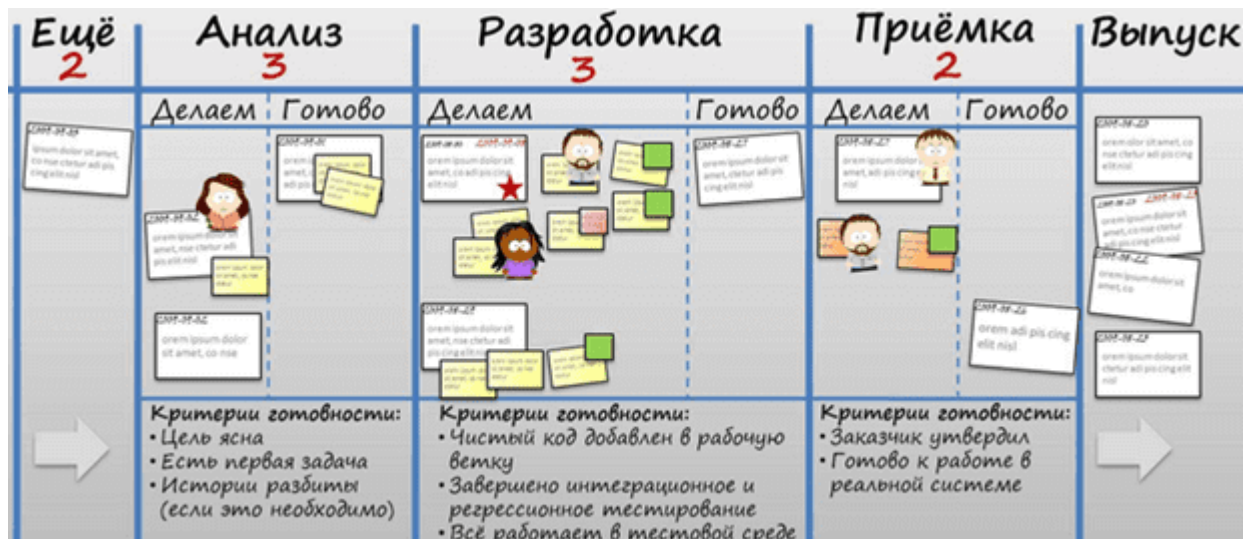


В отличие от Scrum, в команде **канбан** отсутствуют роли владельца продукта и модератора, а процесс разработки делится не на универсальные спринты, а на стадии выполнения задач («Планируется», «Разрабатывается», «Тестируется», «Завершено»). Жизненный цикл задачи отображается на канбан-доске, физической или электронной. Такая визуализация делает рабочий процесс открытым и понятным для всех участников, что особенно важно в Agile, когда у команды нет одного формального руководителя.

### 1.3. Kanban

Канбан, как и другие практики бережливого производства, пришедшие из Японии, направлен на достижение баланса и выравнивание нагрузки исполнителей. Эффективность работы оценивается по среднему времени жизни задачи, от начальной до конечной стадии. Если задача прошла весь путь быстро, то команда проекта работала продуктивно и слаженно. Иначе – необходимо решать проблему: искать, где и почему возникли задержки и чью работу надо оптимизировать



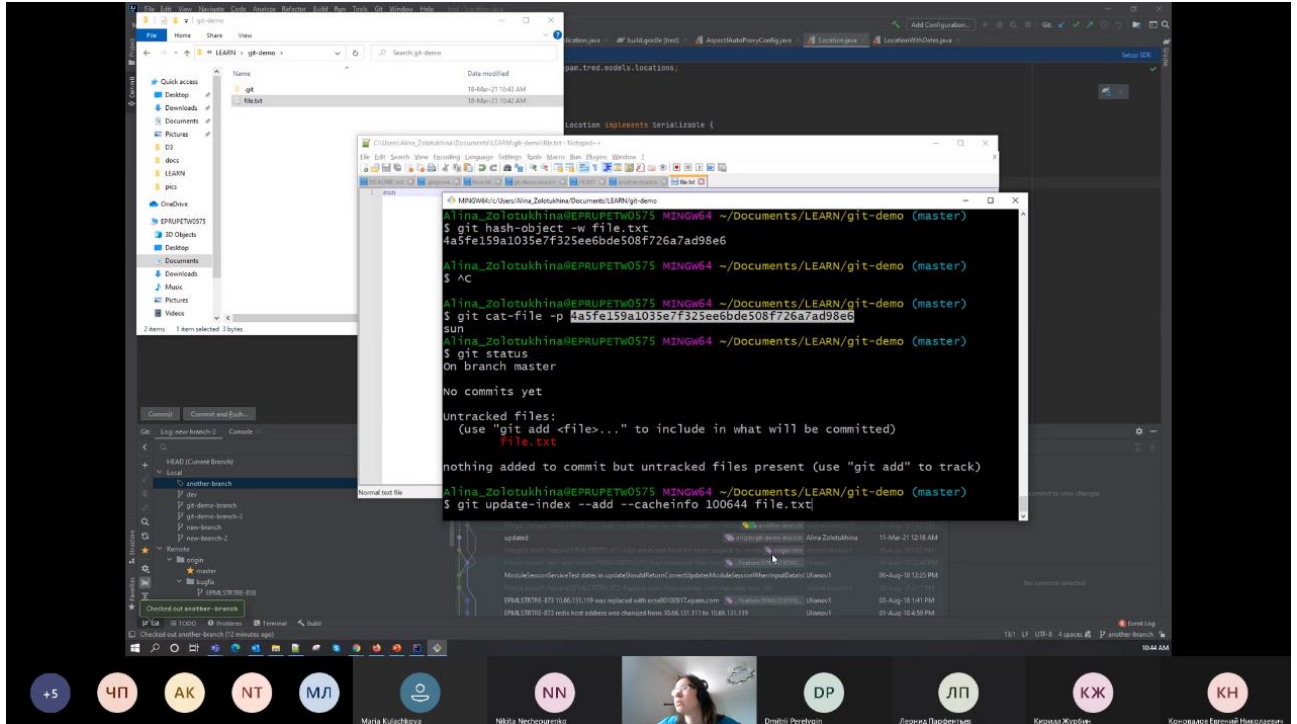


Сегодня наблюдается некоторое слияние Scrum и Kanban, например, канбан-доски стали практически обязательным элементом популярных систем управления проектами (Jira, Trello, Битрикс.24, Basecamp, Мегатлан и т.д.), которые, в том числе, поддерживают методологию скрам.

## 2. Git

Преподаватели: Alina Zolotukhina

Занятий проведено: 2



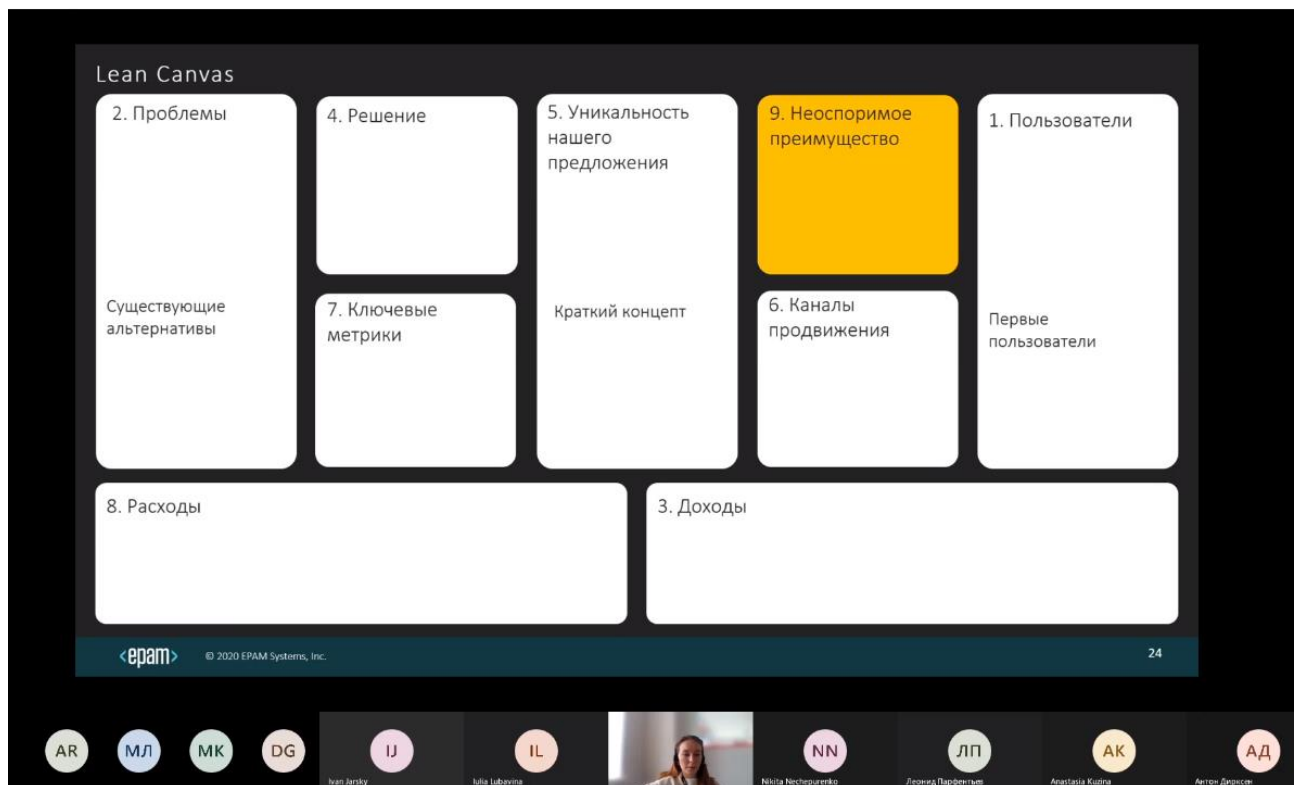
Git - распределённая система управления версиями. Проект был создан Линусом Торвалдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года.

Были рассмотрены основные команды для работы с Git, его внутренне устройство, особенности работы.

### 3. Бизнес-анализ

Преподаватели: Alisa Khoroshenina, Mikhail Ivanov

Занятий проведено: 2



Что такое бизнес-анализ и зачем он нужен?

Бизнес-анализ – это набор методик, которые помогают понять структуру, особенности компании клиента, определить её потребности и предложить варианты решения задачи. В ходе бизнес-анализа выбирают оптимальное решение; готовят к нему требования; оценивают, какая функциональность больше всего важна заказчику; составляют и согласовывают документацию для разработчиков. В IT бизнес-аналитики работают с информационными системами – сайтами и приложениями.

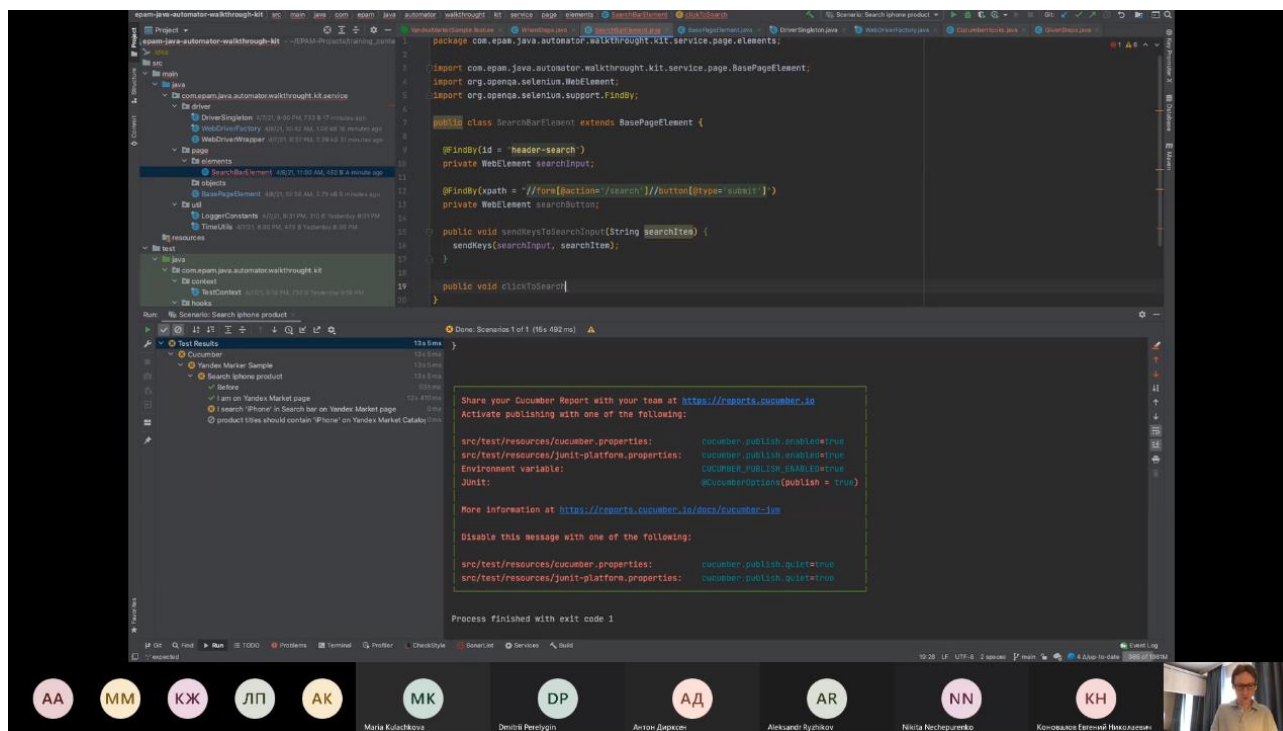
Какие задачи выполняют бизнес-аналитики на проектах?

- Знакомятся с бизнесом заказчика
- Выявляют потребности бизнеса
- Участвуют в разработке требований к системе
- Подготавливают и согласовывают документацию
- Помогают сформулировать задачи разработчикам
- Участвуют в сдаче продукта

## 4. Автоматическое тестирование

Преподаватели: Dmitry Khodakovsky

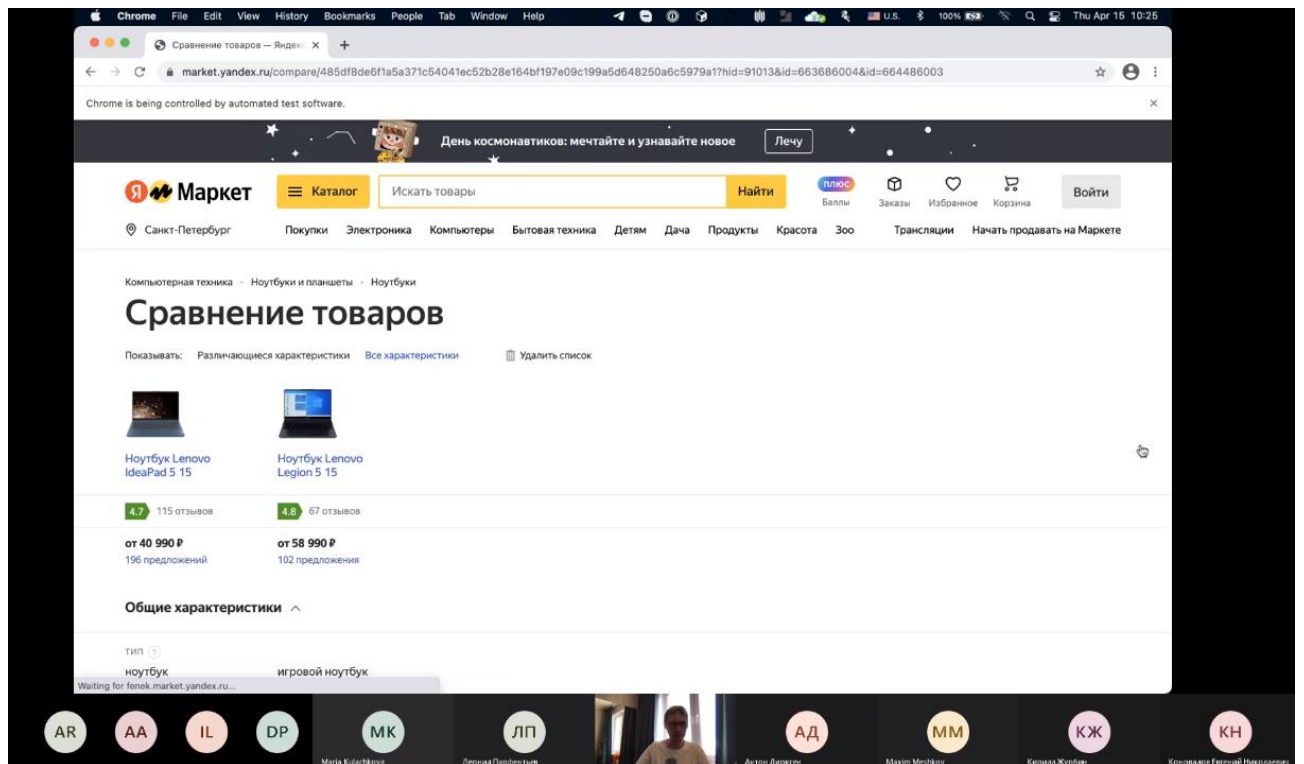
Занятий проведено: 2



### Юнит-тесты

В ходе разработки программисты создают большое количество взаимосвязанного кода и, конечно, делают ошибки. Кто-то больше, кто-то меньше, но пока роботы не научились программировать, никто не застрахован от ошибок. К таким ошибкам относятся опечатки, копирование кода из других источников, программисты могут забыть переименовать какие-то переменные, перепутать названия классов, таблиц, файлов. Чем раньше эти дефекты будут устранены, тем быстрее вы сможете выпустить продукт.

Юнит-тесты создаются для тестирования кода. Объектом тестирования являются классы и методы, разработанные программистом. Преимуществом юнит-тестов является высокая скорость исполнения – как правило весь набор тестов выполняется несколько секунд. Поэтому программисты могут выполнять их так часто, как требуется, например, перед каждым коммитом.



Были разобраны такие инструменты тестирования как:

Cucumber: <https://cucumber.io/docs/gherkin/>

WebDriver: <https://www.selenium.dev/documentation/en/webdriver/>

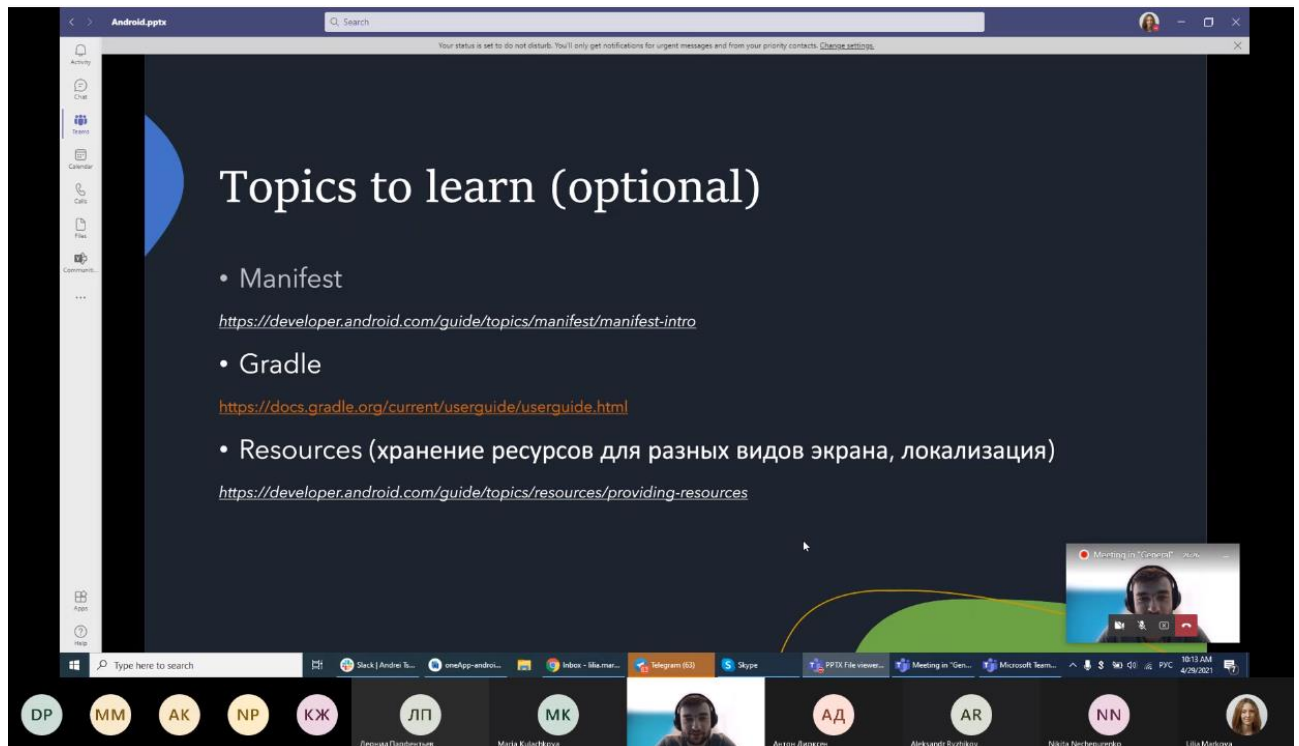
Selenium: <https://kreisfahrer.gitbooks.io/selenium-webdriver/content/index.html>

В качестве демонстрации работы было произведено тестирование сайта Яндекс.Маркет.

## 6. Android

Преподаватели: Lilia Markova, Evgenii Grigoriev

Занятий проведено: 3



Было проведено два вводных занятия по основам android разработки, на 3-ем занятии были представлены реализованные студентами приложения.

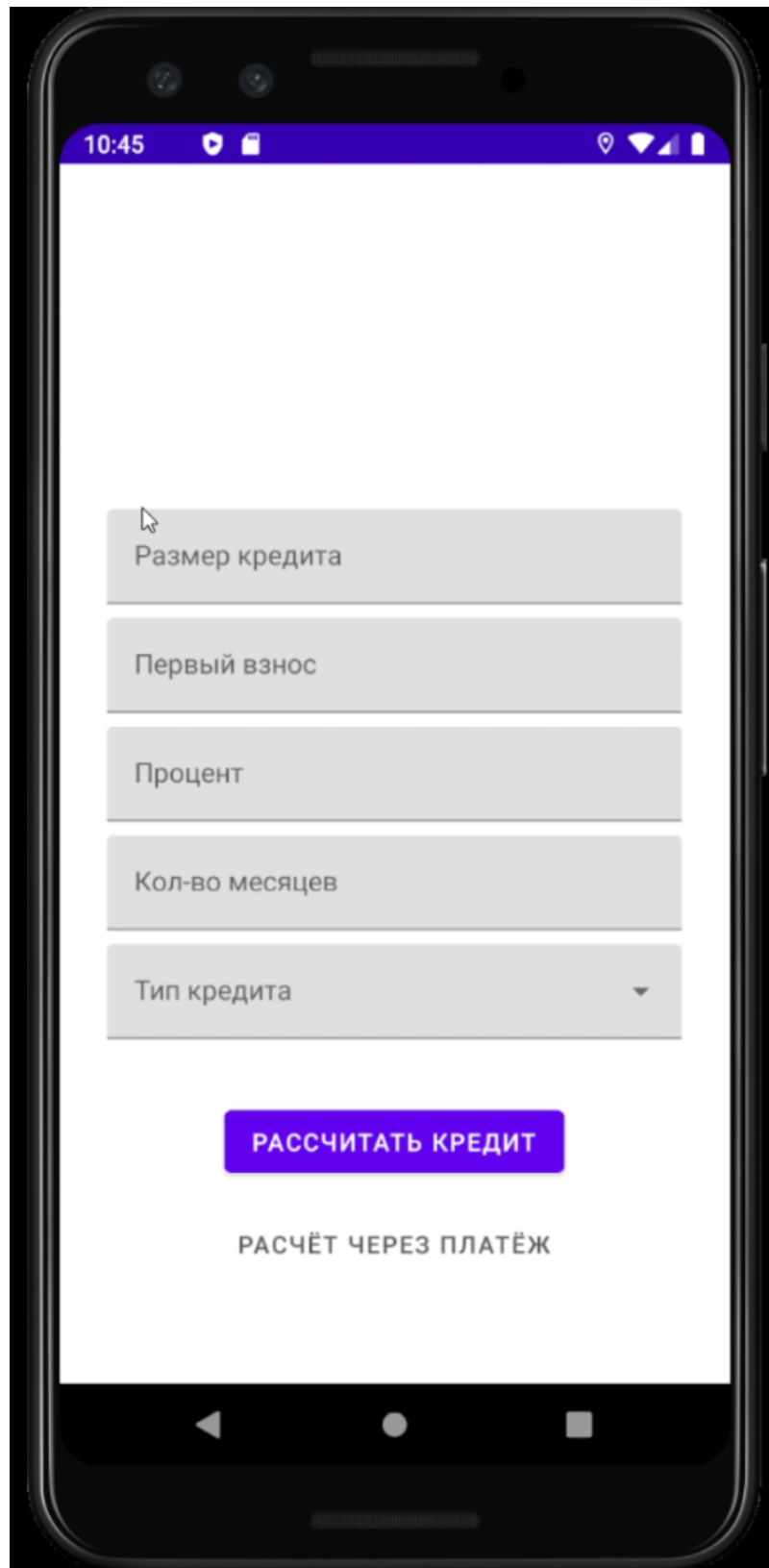
### Реализованное приложение: кредитный калькулятор

Приложение рассчитывает период и размер ежемесячного платежа, составляет график выплат и общую переплату.

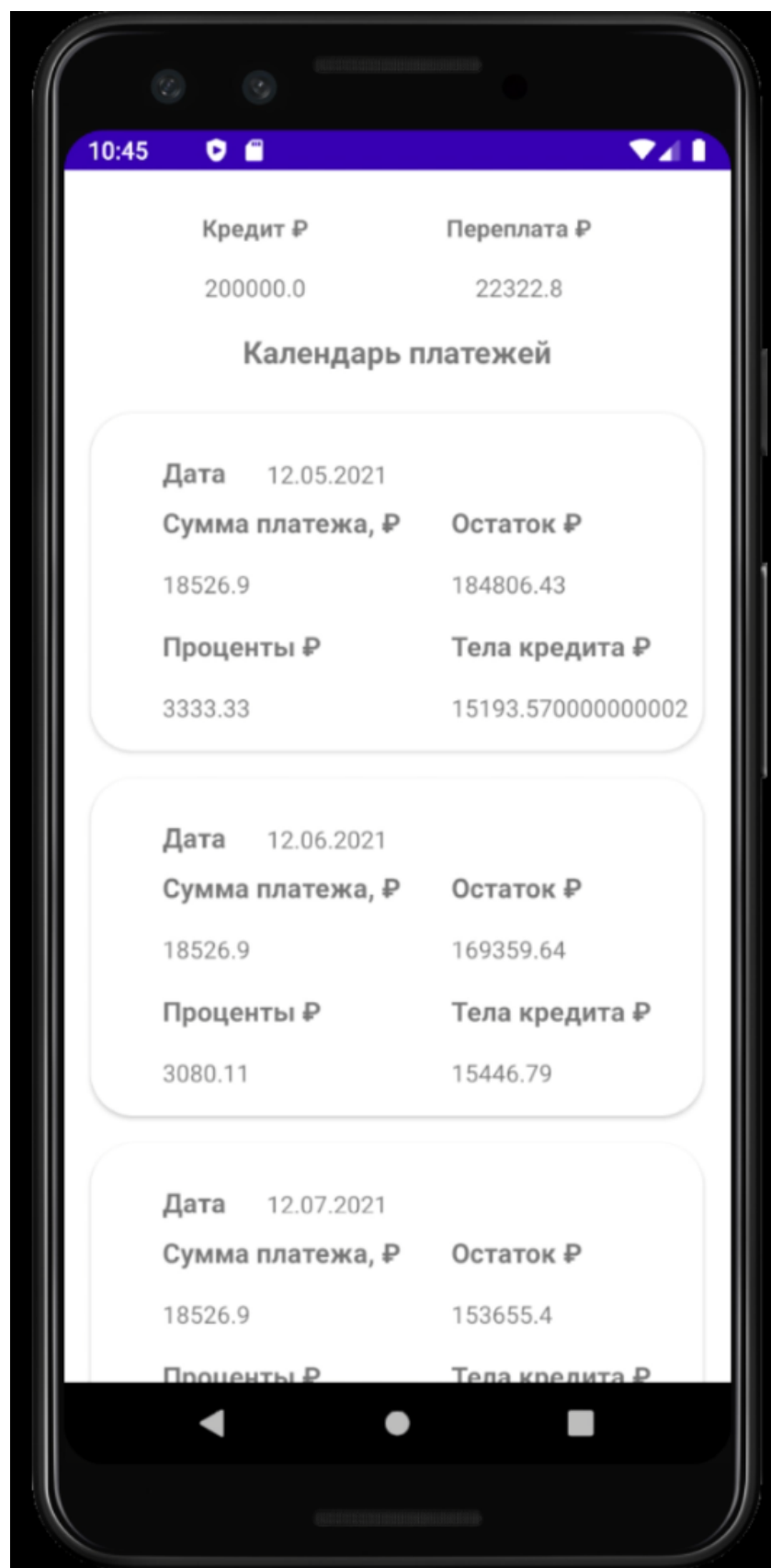
Github: <https://github.com/AlexRyzhickov/Credits-App>

### Демонстрация работы программы

<https://youtu.be/tqfZoGMyW4U>



Экран ввода данных о кредите



Экран с календарём выплат и переплатой



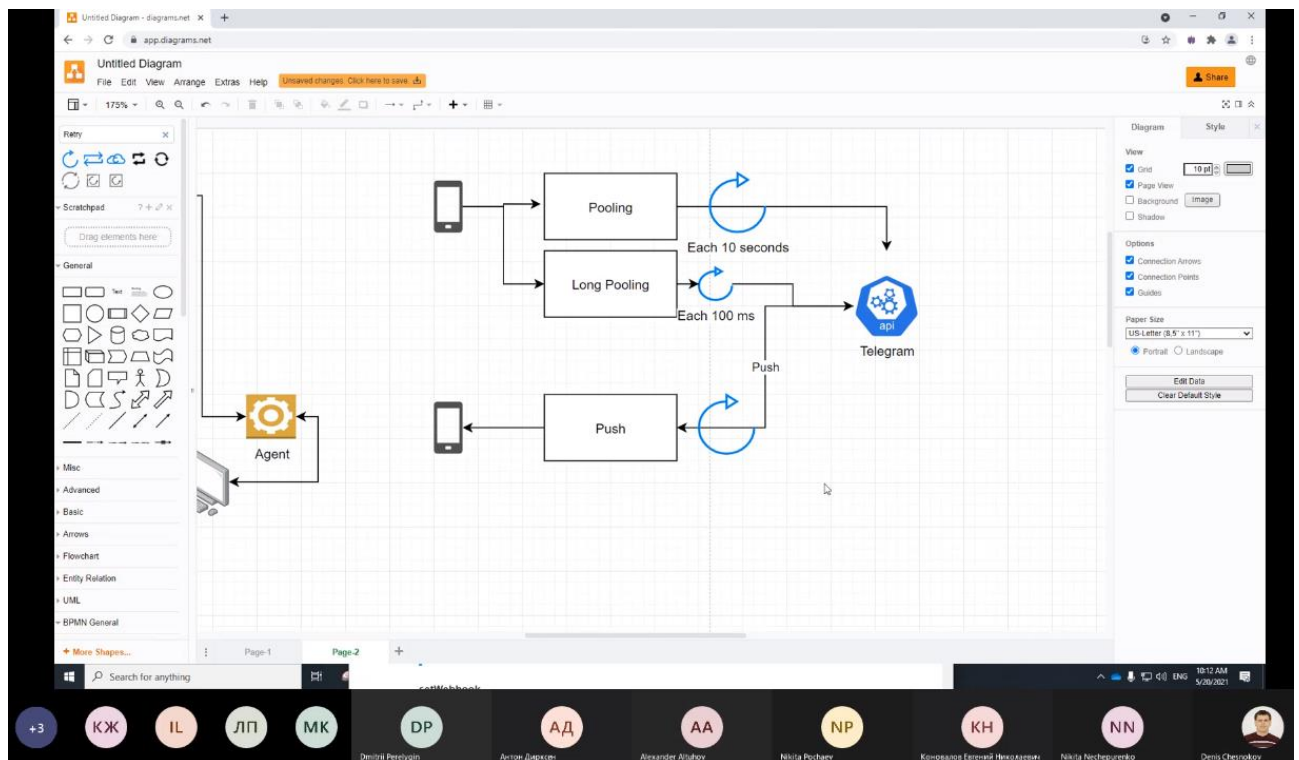
## Материалы для расчёта аннуитетного и дифференцированного порядка погашения

- Погашение аннуитетного платежа <https://temabiz.com/finterminy/ap-formula-i-raschet-annuitetnogo-platezha.html>
- Погашение дифференцированного платежа <https://temabiz.com/finterminy/dp-formula-raschet-differencirovannogo-kredita.html>

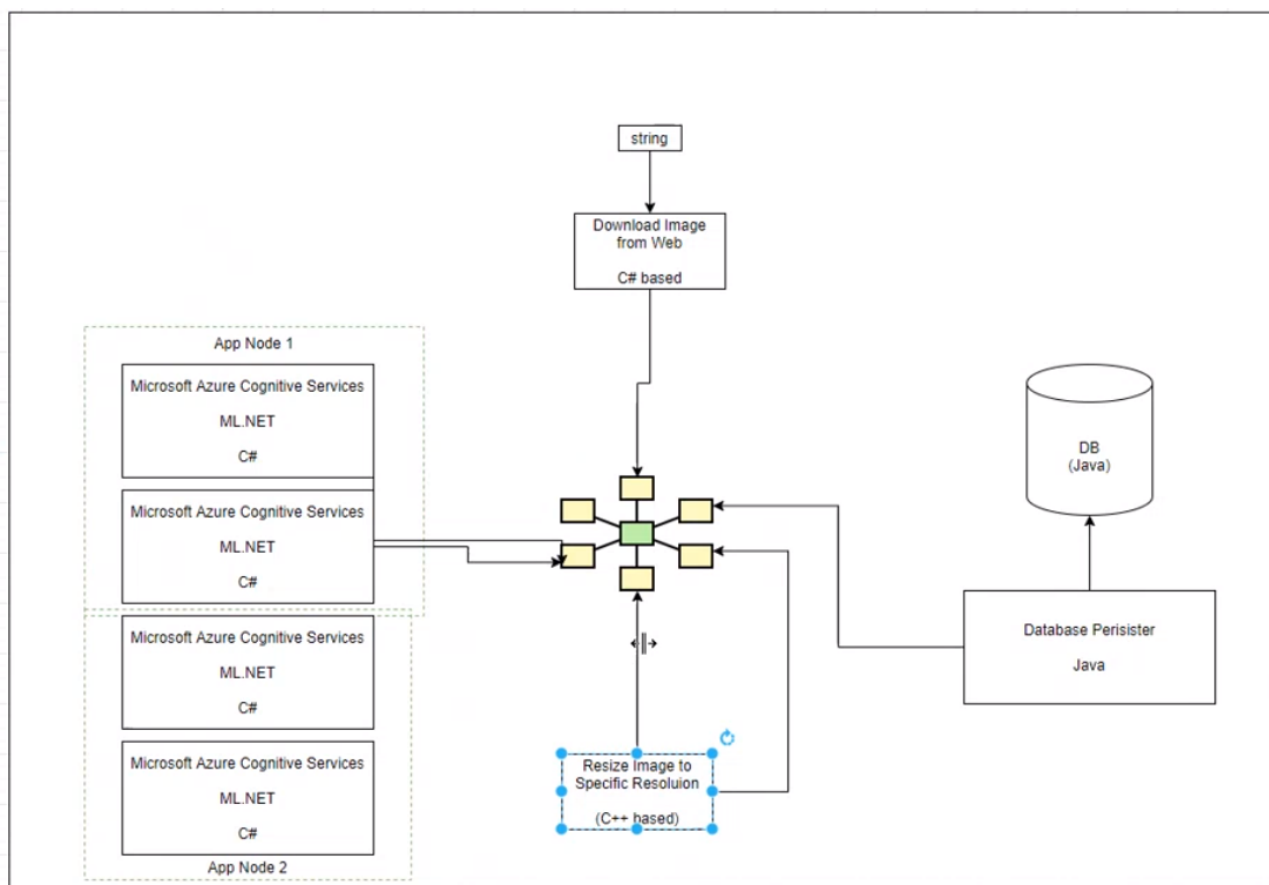
## 7. .NET/C#

Преподаватели: Denis Chesnokov

Занятий проведено: 3



Конечной целью проведения занятий было написание telegram-бота. Для этого были разобраны принципы построения и работы Telegram-API, рассказана основа стека .NET/C#. Также были написаны тесты для тестирования реализованного функционала. Также была затронута тема построения распределённых систем.



Пример распределённой системы

## **ЗАКЛЮЧЕНИЕ**

На практике в компании ерат была изучена agile методология и работа с Git. Также удалось познакомиться с профессиями тестировщика и бизнес-аналитика в реальных условиях IT компании. Завершили практику занятия с android и .NET/C# программистами.