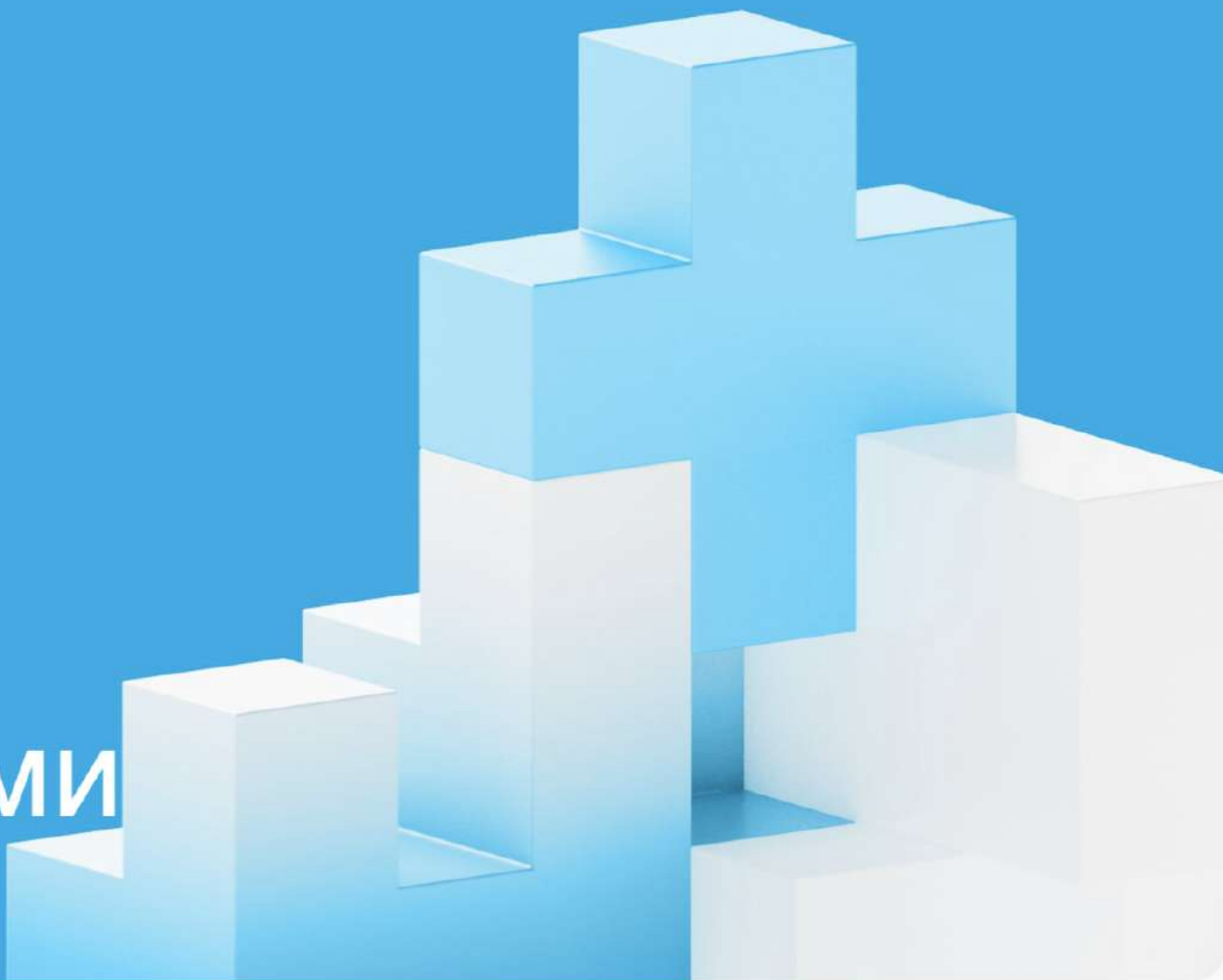


+IT1

Работа с
требованиями



Преподаватель

Белов Иван Иванович + | ТІ Иннотех

Эксперт-аналитик

Образование

Восточно-сибирская академия образования

Факультет компьютерных наук

Прикладная информатика, специалитет

Дискретная математика и математическая кибернетика, аспирантура

Опыт работы



МГЛУ
Московский государственный
лингвистический университет



Содержание

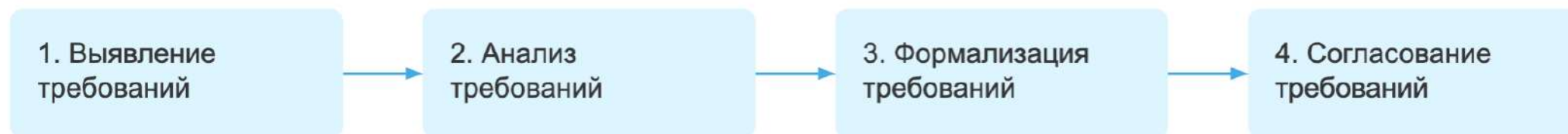


1. Процесс анализа
2. Выявление и анализ требований
3. Основные методы сбора требований
4. Типы требований
5. Качество требований
6. Трассировка требований
7. Формализация и согласование требований
8. Построение User story map

Процесс анализа: работа с требованиями



Цель: получить из бизнес-пожеланий готовую «инструкцию» для разработчика



Выявление и анализ требований



Заказчик всегда просит не то, что он хочет. А хочет не то, что ему нужно.

5 почему / зачем

- n-вопросов, помогающих найти корень проблемы

Дерево проблем

- различие симптомов и проблем
- инструмент как дополнение к «5 почему»

Область выявления проблемы и область решения

- важно разделять этап формулирования проблем и этап формирования решения – часто решение подменяет описание проблемы



Основные методы сбора требований



Какие методы сбора требований вы используете?

+ Интервью

Может быть индивидуальным или групповым, очным или заочным

+ Анализ данных

Изучение документации, работа с существующими версиями ПО и т. д.

+ Анкетирование

Специально оформленный список вопросов, открытого или закрытого типа

+ Работа в «полях»

Непосредственное наблюдение за рабочим процессом

+ Мозговой штурм

Генерация идей небольшой группой, их отбор (фильтрация), расстановка приоритетов

Типы требований



Различаем требования по:

Уровню

1. Бизнес-требования

- Пример: Услуга должна быть легко получена пользователем – быстрое оформление карты
- Проблемы: невыполнимость, невозможность проверить, привязка к конкретному продукту/решению, сомнительная польза

2. Пользовательские требования

- Пользовательские сценарии, User Story и Job Story
- Пример: Хочу иметь возможность пополнять счет через систему СБП для того чтобы иметь возможность переводить средства со счета другого банка

Характеру

1. Функциональные (ФТ)

- Что должна делать система?
- Пример: Автоматически отправлять письмо подтверждения регистрации

2. Нефункциональные (НФТ)

- Как она должна это делать?
- Пример: Отправлять письмо в течении 6ч после регистрации пользователя

Job Stories



Для чего нужны?

Метод описания потребностей пользователя, который фокусируется на контексте, желаемом результате и пользе для пользователя

Структура Job Story:

Когда (состояние) ... хочу ... чтобы ...

Пример

Когда я купил билет на самолет, я хочу получать push-уведомления в приложении о переносе рейса, чтобы приехать на рейс вовремя

Качество требований



2 **Недвусмысленность**
Как НЕ надо: При необходимости пользователь может закрыть карту

3 **Полнота**
Набор требований полный, когда он описывает все необходимые пользователю функции системы

5 **Упорядочиваемость**
Возможность приоритезации разработки

6 **Проверяемость**
Возможность протестировать каждое требование

4 **Непротиворечивость**
Дополнительный уровень защиты для входа в приложение, но такой, чтобы не запоминать код или пароль
Решение: отпечаток пальца / Face ID

7 **Модифицируемость**
Возможность быстро и безболезненно изменять набор требований

8 **Трассируемость**
+ Каждое элементарное требование возможно отследить
+ Требования имеют уникальный ID

Задача 1



Требование

Система должна отправлять уведомление в случае неуспешной регистрации заказа.

Определить

- + Вид требования?
- + Соответствует ли требование критериям качества?

Ответ

Функционально требование

Не соответствует критериям качества

Задача 2



Требование

Приложение должно быть способно обрабатывать запросы одновременно 1000 пользователей.

Определить

- + Вид требования?
- + Соответствует ли требование критериям качества?

Ответ

Нефункциональное требование

Соответствует критериям качества

Задача 3



Требование

При необходимости пользователь может отменить заказ.

Определить

- + Вид требования?
- + Соответствует ли требование критериям качества?

Ответ

Функционально требование

Не соответствует критериям качества

Задача 4



Требование

Платформа должна предоставлять возможность регистрации новых пользователей с указанием имени, фамилии, электронной почты и пароля (поле «Пароль» и «Повторно введите пароль»).

Требование к полю «Имя»:

- Поле должно позволять ввод только букв;
- Поле должно быть обязательным для заполнения;
- Имя должно содержать от 2 до 50 символов;
- Имя не должно содержать пробелов, цифр, специальных символов или знаков препинания (кроме дефиса для двойных имен).

...

Определить

- + Вид требования?
- + Соответствует ли требование критериям качества?

Ответ

Функциональное+Нефункциональное требование

Соответствует критериям качества

Трассировка требований



Зачем нужна?

+ **Управление бэклогом:** можем проверять появляющиеся требования на предмет полезности и приближения нас к цели (сопоставление целей и требований)

+ **Анализ изменений:** как изменения какого-либо требования повлияют на другие требования

Пример влияния новых требований на старые:

Новое \ Старое	Согласовано			
	Велосипед должен ехать	Велосипед должен иметь багажник	Велосипед должен поворачивать направо и налево	Велосипед должен быть легким (менее 5 кг)
Добавить колесо	X		X	X
Добавить детское сиденье		X		X
Добавить встроенный замок				X

Формализация и согласование



- + Формализация
 - + Стил ь написания требований
 - + Формат описания требований
- + Шаблон требований
 - + Шаблон требований на проекте
 - + Требования к ИС
 - + Архитектура ИС
 - + Руководство пользователя
 - + Руководство администратора
 - + Документация для выхода в пром и другие артефакты
 - + Требования к оформлению от заказчика
- + Процесс согласования
 - + Лицо принимающее решение на проекте (при необходимости)
 - + Лицо принимающее решение со стороны заказчика
 - + Разработчики (best practice)



User Story Map



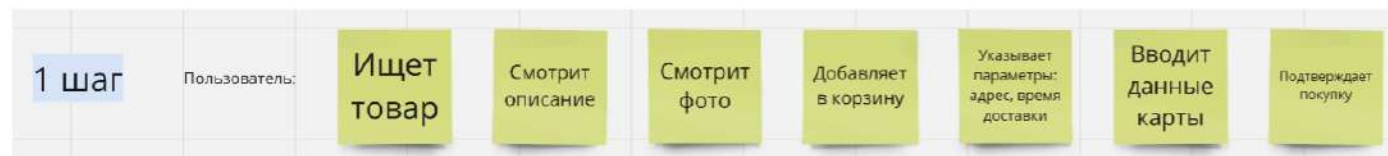
Зачем нужен?

- + Для визуализации клиентского пути, выявления точек, в которых требуется разработка/доработка функциональности
- + А именно: учет пользовательского опыта, выделение MVP, выравнивание в понимании пользовательских процессов на уровне и команды разработки, и заинтересованных лиц
- + Это точно нужно аналитику? - да, важно понимать границы функционального блока, грамотно его декомпозировать и приоритезировать

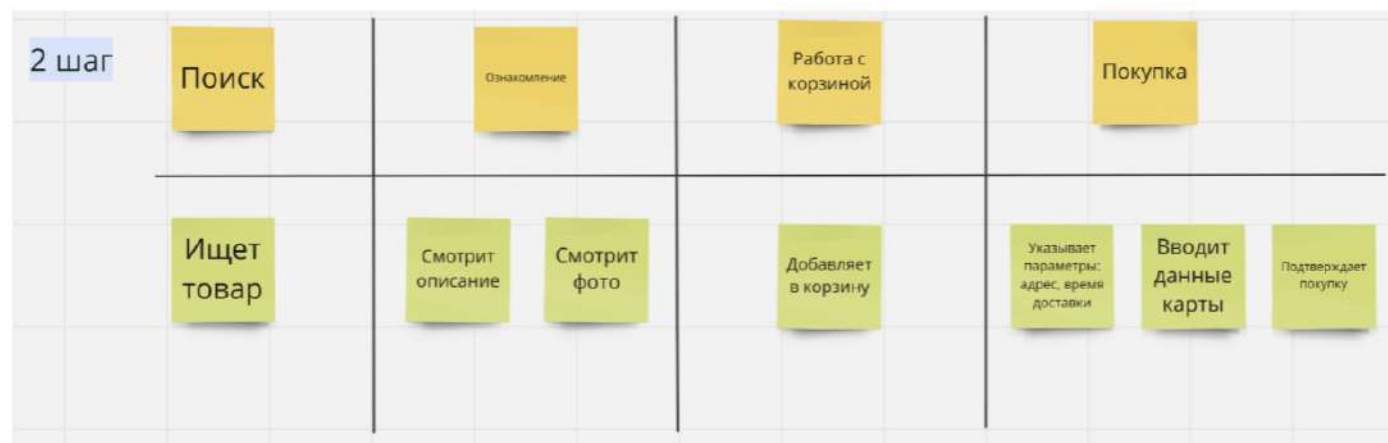
User Story Map



Рассмотрим простой пример: покупка чего угодно в каком угодно Интернет-магазине



Расписываем шаги пользователя: какой путь он совершает обычно, чтобы что-то купить?



Группируем!

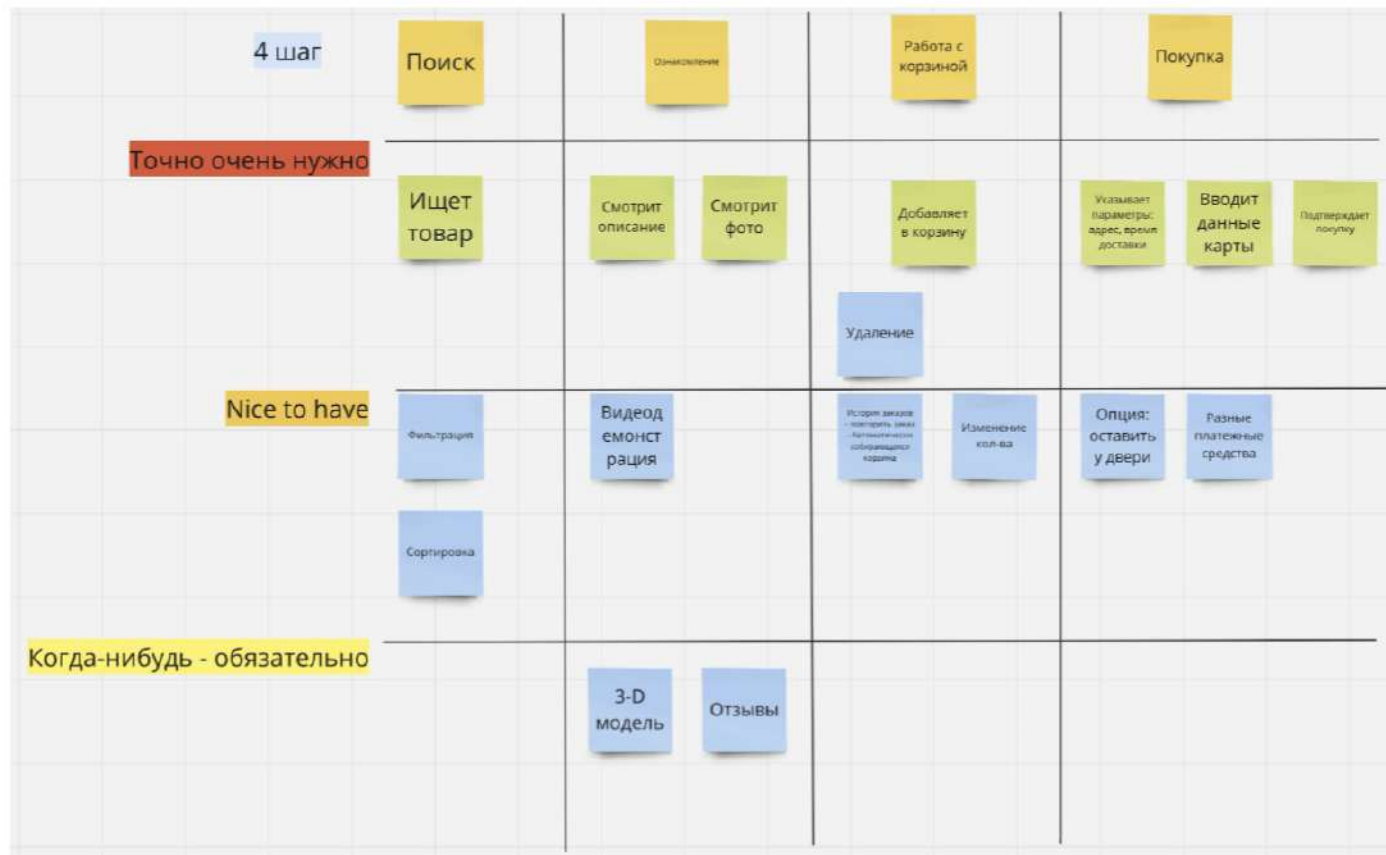
User Story Map



Дополним USM функциями, которых не хватает, например:

- + иметь возможность сортировки и фильтрации при поиске,
- + удаление товара из корзины,
- + выбрать доставку «оставить у двери»

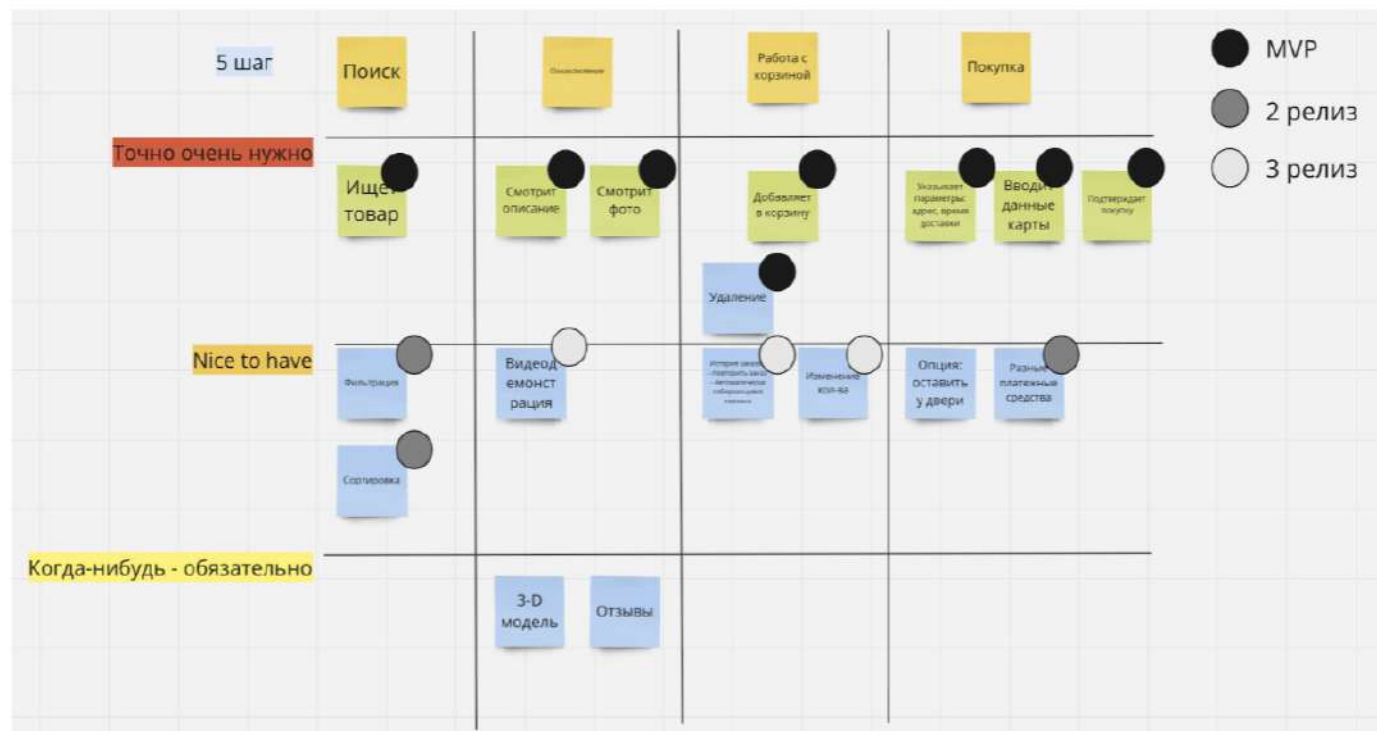
User Story Map



Распределим стикеры USM на 3 категории:

1. **Точно очень нужно** – основная функциональность системы
2. **Nice to have** – фичи, упрощающие жизнь пользователя, но их отсутствие не блокирует работу в системе
3. **Когда-нибудь – обязательно** – ещё менее обязательная функциональность, чем nice to have

User Story Map



Отделим MVP и пару ближайших релизов: остальное будем считать бэклогом

Важно!

Приоритет выполнения стори внутри релиза определяется процессом: мы не реализуем ввод данных карты, пока у нас нет поиска товара 😊

User Story Map



Результаты

- + Пользовательский процесс обеспечен функциональностью
- + Расставлены приоритеты
- + Расставлены приоритеты реализации

Д. 3.



Сдать до 16.02.2025 (включительно)

1. Выбрать предметную область;
2. Описать проблематику и цель (AS-IS и TO-BE);
3. Составить 3-5 и более User Story;
4. Сформулировать 3-5 нефункциональных требования;
5. Сформулировать 3-5 функциональных требования;
6. Выбрать 3 критерия качества и написать на каждый по «плохому» требованию (для которого этот критерий не выполняется);
7. Сформировать User Story Map, выделив MVP, следующий релиз и бэклог (как инструмент можно использовать <https://miro.com/>).