Практические занятия 5-8

ДИСЦИПЛИНА	Технологии управления командами разработчиков программного обеспечения
ИНСТИТУТ	Институт перспективных технологий и индустриального программирования
КАФЕДРА	Кафедра индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Практические занятия
ПРЕПОДАВАТЕЛЬ	Зарипова Виктория Мадияровна
СЕМЕСТР	3 семестр, 2025-2026 гг.

Тема

Составление WBS

Цель занятий

Практика по организации рабочих встреч и уточнению скопа задач с командой Заказчика.

План занятий

- Освоить метод Event Storming
- Освоить метод User Story Map

Входные знания

- Необходимы базовые знания о User Story.
- Необходимо иметь понимание состава персонажей, задействованных в ПО
- Необходимо изучить продукты аналоги и состав их функций

Итоги работы

Студенты научатся взаимодействовать с заказчиком по выстраиваиванию логики работы приложения через Event Storming и User Story Map, применять стратегии Ubiquitous Language, Bounded Context, Shared Kernel.

Научатся нарезать scope проекта на блоки

Практические итоги работы и отчетные материалы

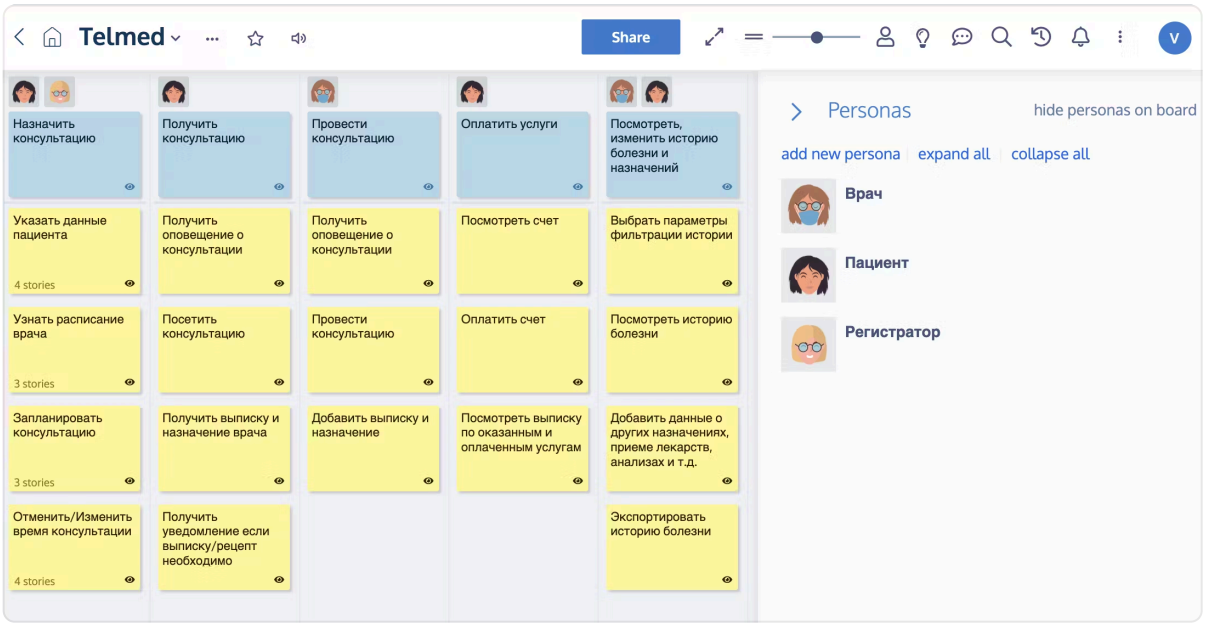
- Карта Event Storming (как минимум события, команды, агрегаты и внешние системы)
- Карта User Story Map (как минимум на MVP)
- Матрица WBS в https://app.clickup.com/

Справочный материал:

- См. лекции

Карта пользовательских историй проекта

С помощью инструмента Stories On Board разработайте карту пользовательских историй проекта



Цели и шаги пользователей по бизнес-процессу

Роли	Цель	Шаги
▼ Пациент, Оператор 4	▼ Назначить консультацию 4	Указать данные пациента
		Узнать расписание врача
		Запланировать консультацию
		Отменить/Изменить время консультации
▼ Пациент 5	▼ Получить консультацию 4	Получить оповещение о консультации
		Посетить консультацию
		Получить выписку и назначение врача
		Получить уведомление или выписку/рецепт
	▼ Оплатить счет 1	Посмотреть счет Оплатить счет
▼ Врач 3	▼ Провести консультацию 3	Получить оповещение о консультации
		Провести консультацию
		Добавить выписку и назначение

Описание шагов по требованиям заказчика

Пользовательская история №1: Назначить консультацию

Шаг 1.1: Указать данные пациента
Пациент заходит на сайт и регистрируется, создавая персональный профиль. В процессе регистрации, пациент обязан заполнить поле с контактной информацией и основными данными, а именно, ФИО, возраст, адрес электронной почты и номер телефона.

Шаг 1.2: Узнать расписание врача
MVP: Чтобы узнать расписание врача, пациенту нужно перейти в раздел "Консультации" на сайте и выбрать в меню нутрициологию. Здесь указаны все доступные врачи с подробным расписанием и указанием свободных и занятых слотов.

v1.2: В интерфейсе пользователя предусмотрен раздел "Врачи и расписание". Пациенты могут просмотреть полное расписание всех врачей, либо использовать сортировку и фильтры для выбора конкретного специалиста.

Пациент может воспользоваться следующими опциями поиска:

- По ФИО врача: пациент вводит имя в строку поиска, и система отображает специалистов с совпадающими данными и их расписание.
- По специализации: в списке с разделами на сайте есть пункт "Специализации". Пациенту предоставляется возможность выбрать интересующую его область

Шаг 1.3: Запланировать консультацию
После выбора подходящего врача и времени, пациент нажимает кнопку "Записаться на консультацию". Запись подтверждается автоматически и сохраняется в персональной карточке пациента.

Шаг 1.4: Отменить/Изменить время консультации
В случае, если пациенту нужно изменить или отменить время консультации, он может сделать это через свой персональный профиль, выбрав соответствующую опцию в разделе "Запланированные консультации".

Пользовательская история №2: Получить консультацию

Шаг 2.1: Получить оповещение о консультации
За сутки до консультации пациенту будет отправлено уведомление об актуальности его записи на медицинскую консультацию. Оповещение может быть отправлено по электронной почте или SMS-сообщением.

Шаг 2.2: Посетить консультацию
В указанное время, пациенту необходимо зайти в свой аккаунт и перейти в раздел "Мои консультации". В указанное время встречи со специалистом откроется окно видеоконференции.

Шаг 2.3: Получить выписку и назначение врача
После консультации врач заполняет выписку с описанием состояния здоровья пациента и дает рекомендации. Эту выписку пациент может найти в разделе "Мои консультации".

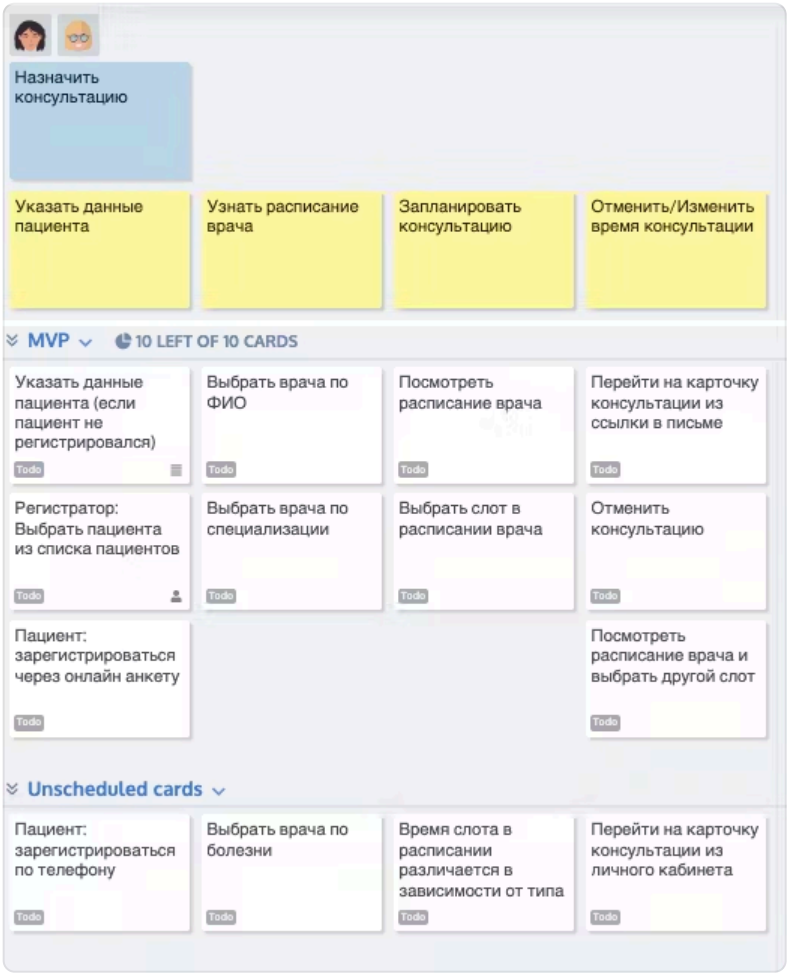
Шаг 2.4: Получить уведомление или выписку/рецепт
Если врач выписал рецепт, пациент получает об этом уведомление по электронной почте, а в разделе "Мои консультации" пациента будет прикреплен собственно рецепт в электронном виде.

Пользовательская история №3: Оплатить счет

Шаг 3.1: Посмотреть счет
Для просмотра счета пациенту нужно зайти в раздел "Оплата" своего личного кабинета. Здесь представлены все неоплаченные счета.

Шаг 3.2: Оплатить счет
Пациент может оплатить консультацию напрямую через сайт. Для этого достаточно выбрать соответствующий счет, указать способ оплаты или ввести данные банковской карты, и подтвердить платеж.

Для каждого шага укажите какой функциональностью он будет закрыт на каждом этапе проекта



DDD и Event Storming

DDD (Domain Driven Design)

Доменно-ориентированное проектирование (Domain-Driven Design, DDD) - это подход к разработке программного обеспечения, при котором внимание уделяется бизнес-логике и комплексности бизнес-процессов.

В контексте DDD, "домен" представляет собой сферу знаний и активности, в которой применяется разрабатываемое программное обеспечение. Домен может быть любым бизнес-процессом или системы знаний конкретной области.

Тезисы DDD:

1. Ориентирован на моделирование сложных бизнес-процессов.
2. Необходим плотный контакт с экспертами по предметной области.
3. Во всех контекстах проекта (документация, API, соглашения о наименованиях в коде, UI, отчетность) используется язык предметной области (Ubiquitous Language)
4. Сложность проектирования, разработки и поддержки управляется с использованием стратификации - разделения логики проекта на максимально независимые представления
5. Эволюция проекта через постоянный рефакторинг моделей

Стратификация

Стратификация в контексте DDD — это разбиение программного обеспечения на слои или группы, каждый из которых имеет свое предназначение и ответственность. Это помогает управлять сложностью проекта, позволяя команде разработки фокусироваться на отдельных частях системы независимо друг от друга.

Обычная модель стратификации включает в себя несколько слоев, таких как слой представления (UI), слой приложения (оркестрирует бизнес-логику и делает всю грязную работу), слой домена (где живет вся бизнес-логика), и слой инфраструктуры (обеспечивает поддержку для других слоев, включая сетевое взаимодействие, доступ к базе данных и т.д.).

Такое разделение позволяет изолировать и абстрагировать разные части системы, снижая сложность проекта и делая его управление более эффективным.

Ubiquitous Language

Эксперты заказчика могут объяснять бизнес-процессы одним образом, а технические эксперты могут их иначе интерпретировать. Этот процесс может привести к созданию технических моделей (возможно, в UML), которые эксперты по предметной области могут не понять.

Таким образом, должна быть сессия, на которой обе стороны обмениваются знаниями на одном языке. Это позволит требованиям выявиться естественным образом и обеим сторонам, бизнес и техническим, легко создать модель предметной области.

В контексте DDD, этот "общий язык" часто называется "всеобщим языком" (Ubiquitous Language), и его использование помогает гарантировать, что все участники проекта имеют общее понимание домена и требований.

Event Storming

Event Storming - эффективный метод моделирования сложных бизнес-процессов, который используется в рамках DDD. Он позволяет сконцентрироваться на бизнес-логике и очень хорошо подходит для работы с большими и сложными системами.

Вот некоторые ключевые тезисы:

1. Совместная работа: Event Storming проводится в группе, которая включает как технических специалистов, так и экспертов по предметной области (домен).
2. Визуализация: используются цветные стикеры для записи и визуализации концепций, событий, команд и многого другого, что связано с бизнес-процессом. Это очень важно для общего понимания предметной области.
3. Исследование бизнес-процессов: это подход, позволяющий быстро выявлять ключевые аспекты бизнес-логики и области, которые могут представлять проблемы или быть недопонятыми.
4. Обучение: этот метод позволяет каждому участнику лучше понять предметную область и обеспечивает пространство для обсуждения и решения проблем.
5. Результаты: на выходе получается визуальная модель бизнес-процесса, которая может быть использована как основа для проектирования и разработки системы.

С помощью Event Storming команды могут быстро создать общее понимание сложного домена, идентифицировать возможные риски и проблемы, а также формировать базис для более детального моделирования и реализации системы.

Требования к проведению сессии Event Storming

- Среда или окружение:
  - Минимум 8 цветов стикеров. Есть стандартные цвета, которые используются обычно, но можно выбрать свои, главное дать карту используемых цветов
  - Текст на карточке должен быть хорошо читаемым и коротким, поэтому для стенда используйте маркеры, для виртуальной доски не мельчите шрифт.
  - Используйте любую физическую или виртуальную стену на которую вы можете наклеить карточки. Помните о том что вам нужно зафиксировать результаты сессии, так что либо наклейте на стену бумажные листы, которые потом можно аккуратно собрать и унести, либо эта стена останется с вами надолго.
  - Если это встреча в реале, то используйте просторную комнату с хорошим освещением. Помните что уровень шума может быть достаточно большим. Если это виртуальная встреча используйте любое приложение для совместной работы, например:
    - https://miro.com/miroverse/event-storming/,
    - https://www.figma.com/templates/event-storming-example/,
    - https://link.excalidraw.com/l/AqkGvKXqNB0/2IVK7Hq7HTQ - нужно скачать шаблон, создать собственную доску и загрузить шаблон
  - По окончанию встречи зафиксируйте итоги в коротком протоколе

В контексте Event Storming участники имеют следующие роли:

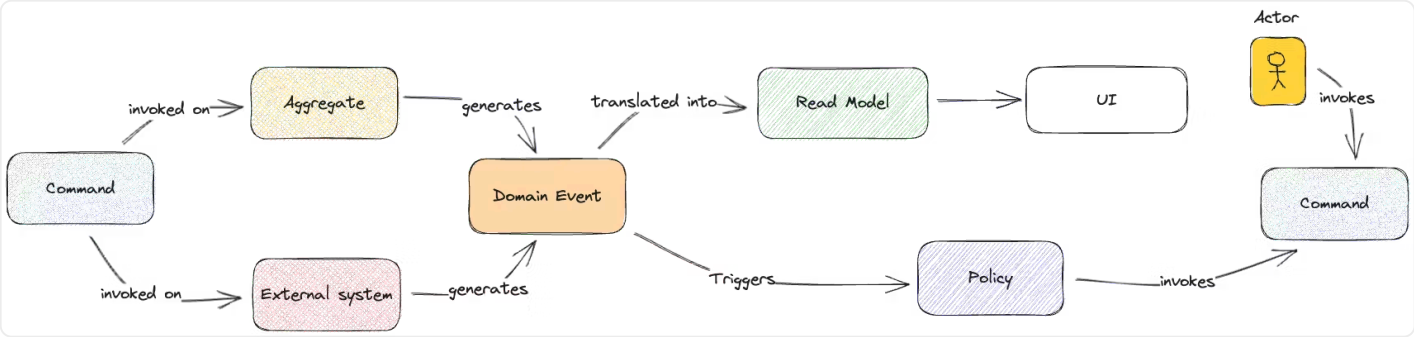
- 1. Фасилитатор (Facilitator): Это лицо, отвечающее за организацию сессии. Фасилитатор необходим для того, чтобы обеспечивать следование правилам Event Storming на протяжении всей сессии. Этот человек задает направление обсуждения, помогает в контроле времени и занимается управлением конфликтами, если таковые возникают.
- 2. Эксперты по домену (Domain Experts): Это люди, представляющие различные поддомены или области бизнеса. Их роль заключается в передаче знаний о бизнес-процессах и функциональности, которые планируется моделировать.
- 3. Технические эксперты (Technical Experts): Это люди, ответственные за проектирование и разработку программного обеспечения. Они должны обладать навыками извлечения требований от экспертов по предметной области, а также преобразования этих требований в архитектуру и дизайн системы.

Общим вкладом каждой из этих ролей является создание совместной модели предметной области и обмен знаниями и

пониманием между участниками сессии.

Легенда цветов и схема

Здесь представлена типовая схема взаимодействия и цветов.



Рекомендуемый порядок действий

1. Сначала укажите все события которые могут произойти в рамках процесса. Можно к примеру указать начальное событие, конечное событие и далее начинать заполнять от события которое приведет к конечному событию.
2. Затем перед каждым событием добавьте команду
3. В случае если у вас намечается вилка значит скорее всего там стоит бизнес-правило или политика, которая определяет дальнейший порядок действий. Политика всегда ставится после ключевого события, которое ее запускает и может вызывать команды.
4. Затем укажите внешние сервисы, вопросы и персонажей
5. Укажите агрегаты и модели чтения
6. После того как все шаги выполнены, необходимо сгруппировать агрегаты вместе чтобы найти границы между функциональностями. Идея здесь в том чтобы обрисовать границей каждый агрегат. Здесь уже не нужно следовать временной последовательности событий.

Смотрите пример здесь: https://link.excalidraw.com/l/AqkGvKXqNB0/DR8I7Mto0s

Bounded Context

Идея заключается в отображении предметной области и выделении в ней ограниченных контекстов. Каждый такой контекст содержит свою модель и язык, который оптимальны для работы в данном контексте. Модель в одном контексте может отличаться от модели в другом контексте. Это помогает обрабатывать сложность бизнес-логики, изолируя разные аспекты системы и упрощая работу над каждым из них отдельно.

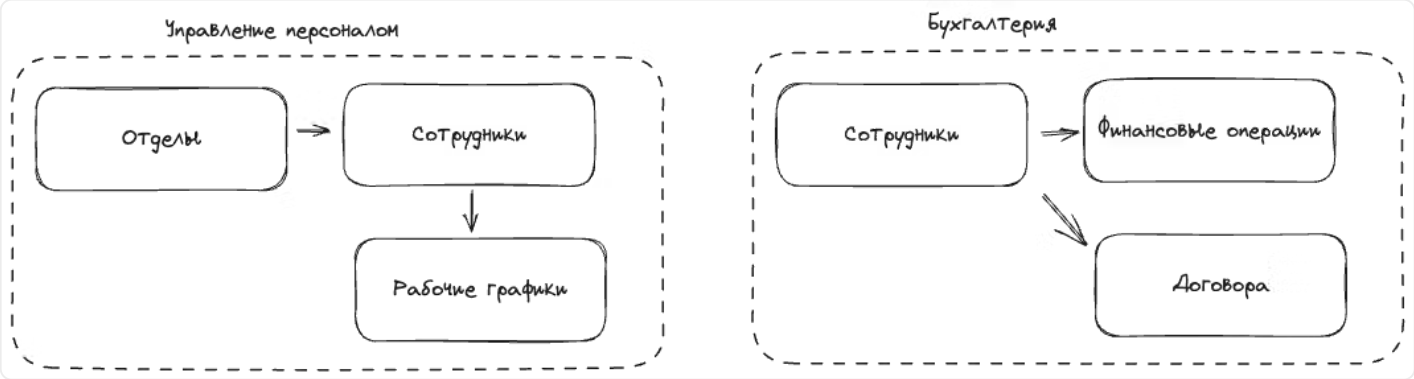
Допустим, у нас есть два домена: "Управление персоналом" и "Бухгалтерия". В обоих этих доменах может быть сущность "Сотрудник".

В контексте "Управления персоналом", "Сотрудник" может включать данные о навыках, перфомансе, должности, обязанностях и т.д. Важной операцией в этом контексте может быть, например, перевод сотрудника из одного подразделения в другое.

В контексте "Бухгалтерия", "Сотрудник" может включать информацию о зарплате, налогах, выплатам и т.д. Здесь важной операцией может быть выплата зарплаты.

Несмотря на то, что в обоих контекстах мы говорим о "сотруднике", эти две модели "сотрудника" довольно разные и используются для разных целей. Важные детали и операции, которые необходимы в одном контексте, могут быть нерелевантны или даже вводить в заблуждение в другом.

Bounded Context в этом примере помогает разделить эти две разные области применения "сотрудника" на два отдельных контекста, каждый со своей собственной моделью и терминологией. Это помогает избегать путаницы и ошибок, а также позволяет при необходимости разрабатывать и эволюционировать эти модели независимо друг от друга.



Shared Kernel

Это подмножество доменной модели, которое может быть разделено между несколькими командами или проектами. Обычно, он содержит основные бизнес-правила и сущности, которые не изменяются часто и должны быть согласованы между разными доменами. Паттерн Shared Kernel способствует сокращению дублирования кода и уменьшает времени на разработку.

"Shared Kernel" в нашем примере с "Управлением персоналом" и "Бухгалтерией" мог бы включать базовую информацию про "Сотрудника".

Эта базовая информация могла бы охватить поля, которые общи для обоих контекстов и доменов. Например, идентификатор сотрудника, имя, фамилия, контактная информация и т.д, то есть все что есть в договоре. Это та информация, которая, скорее всего, не будет меняться часто и должна быть согласована между этими двумя доменами.

Таким образом, "Shared Kernel" помогает уменьшить дублирование и поддерживать общую консистентность данных о сотрудниках на протяжении всей системы.