**Задача: Разработка на софтуерно приложение с методология Scrum**

**Описание на задачата:**

Трябва да разработите ново софтуерно приложение за управление на проекти. Вашата задача е да планирате, организирате и осъществите етапите на разработка, като следвате принципите и процесите на методологията Scrum. Крайният продукт трябва да улеснява екипи в управлението на задачи, разпределението на работата и проследяването на напредъка.

**Изисквания:**

1. **Създаване на backlog (списък със задачи)**:
   * Съберете всички функционални и нефункционални изисквания, зададени от клиента.
   * Подгответе основен списък на потребителските истории, като ги подредите според важността им.
2. **Създаване на Scrum екип**:
   * Определете ролите в екипа (Scrum Master, Product Owner, Developers).
   * Организирайте комуникационните канали между членовете на екипа.
3. **Етапи на разработка и спринтове**:
   * Разделете проекта на спринтове (обикновено 1-2 седмици всеки).
   * Всеки спринт трябва да има конкретни цели и дефиниран продукт, който трябва да бъде завършен и готов за демонстрация.
4. **Разпределение на задачите**:
   * В началото на всеки спринт провеждайте Sprint Planning среща, на която да определите задачите за този спринт и кой ще отговаря за всяка от тях.
   * Следвайте приоритетите, зададени в backlog, и ги адаптирайте спрямо обратната връзка от клиента.
5. **Ежедневни срещи (Daily Stand-ups)**:
   * Провеждайте кратка среща (до 15 минути) всеки ден, за да обсъдите напредъка, предизвикателствата и целите за деня.
   * Следете дали всички работят в съответствие с плана и коригирайте задачите при нужда.
6. **Преглед на спринта (Sprint Review)**:
   * В края на всеки спринт провеждайте Sprint Review, на която демонстрирате постигнатия резултат пред клиента и събирате обратна връзка.
   * Ако клиентът има нови изисквания, ги добавете в backlog.
7. **Анализ на спринта (Sprint Retrospective)**:
   * След Sprint Review направете анализ на постигнатото и преглед на процесите.
   * Открийте какви подобрения може да се приложат в следващия спринт и вземете решения за оптимизиране на процеса.

**Критерии за успех:**

* Приложението да отговаря на зададените изисквания и да бъде функционално и стабилно.
* Всеки спринт да завършва с работещ продукт (доставяема част) и внедрена обратна връзка от клиента.
* Спазване на зададените срокове за всеки спринт.
* Добра комуникация и сътрудничество в екипа, както и своевременно разрешаване на възникнали проблеми.

**1. Създайте основни колони (листи) в Trello**

1. **Backlog**:
   * В тази колона поставете всички задачи и потребителски истории от проекта, които предстои да бъдат изпълнени. Те са събрани в началото на проекта, но ще се преразглеждат и пренареждат в началото на всеки спринт.
   * За по-голяма яснота, всяка задача може да има описание и прикачени файлове с детайлни изисквания.
2. **To Do (за текущия спринт)**:
   * Прехвърлете тук задачите, които ще бъдат изпълнени през текущия спринт, след като са приоритетизирани на Sprint Planning срещата.
   * Всяка задача трябва да има ясно заглавие, описание, критерии за приемане, и приоритет. Може да добавите етикети (labels) за различните видове задачи като "Фронтенд", "Бекенд", "Тестове" и т.н.
3. **In Progress (В процес)**:
   * Тук членовете на екипа премествайте задачите, по които работят в момента.
   * Всяка задача може да бъде асоциирана с конкретен член на екипа чрез опцията за добавяне на членове (members) и определена дата за завършване (due date).
   * Можете да използвате и Checklist за всяка задача, за да следите по-малки стъпки в изпълнението.
4. **Testing (Тестване)**:
   * В тази колона премествайте задачите, които са завършени, но се нуждаят от тестване.
   * Добавете чеклист с всички критерии за приемане, които трябва да бъдат изпълнени за успешното преминаване на тестването.
   * Членовете на екипа, отговорни за тестването, могат да преглеждат и маркират завършените под-задачи.
5. **Done (Готово)**:
   * Тази колона е за задачите, които са завършени и успешно преминали тестване.
   * Задачите се преминават тук след одобрение от Product Owner-а или при успешна демонстрация на клиента.
6. **Sprint Review**:
   * На края на всеки спринт създайте картичка, която описва основните успехи и предизвикателства за съответния спринт.
   * Събирайте обратна връзка от клиента, която може да се добави като нови задачи в Backlog.
7. **Sprint Retrospective**:
   * Създайте картичка за всяка ретроспективна среща, на която екипът да записва какво е било добре и какво може да бъде подобрено.
   * Планирайте нови идеи за оптимизация и ги поставете като задачи за следващите спринтове.

**2. Организиране на задачите и добавяне на детайли**

* **Labels (Етикети)** – Използвайте етикети за различни типове задачи и приоритети (напр. "Висок приоритет", "Фронтенд", "Бекенд", "Блокирано").
* **Members (Членове)** – Присвоявайте задачите към конкретни членове на екипа, за да е ясно кой отговаря за тях.
* **Due Date (Краен срок)** – Определяйте крайни срокове за всяка задача, за да следите времето и напредъка по плана на спринта.
* **Checklist (Списък със задачи)** – При по-сложни задачи добавете под-задачи за ясно проследяване на изпълнението.

**3. Ежедневни stand-up срещи и актуализация на задачите в Trello**

* Всеки ден екипът преглежда напредъка и премества задачите спрямо актуалния статус в съответната колона.
* Обсъждат се евентуални пречки (блокери) и се отбелязват с етикет "Блокирано", за да се разрешат бързо.

С този метод на разпределение в Trello вашият екип ще може лесно да следи напредъка по проекта, като визуализира и управлява задачите в съответствие със Scrum процесите.

Ето как бихте могли да структурирате задачите в Trello, ако проектът ви е уеб приложение за управление и резервации на морски хотели. Ще използваме основните Scrum принципи и ще разпределим задачите в списъци според етапите на разработка:

**Структура на Trello борда за уеб приложението „Морски хотели“**

**1. Backlog (Списък със задачи)**

* В Backlog-а добавете потребителски истории и основните функционалности на приложението. Ето някои примерни задачи:
  + **Регистрация и вход на потребители** – потребителите трябва да могат да се регистрират и влизат в системата.
  + **Търсене на хотел** – филтри по локация, дата, тип на стаята, цена и удобства.
  + **Резервация на стая** – избор на дати и резервация в реално време с потвърждение по имейл.
  + **Преглед на профил на хотела** – страница с детайли за хотела, галерия, описание, удобства и ревюта.
  + **Управление на резервации** – панел за хотелиерите за управление на наличности, цени и резервации.
  + **Отзиви и рейтинги** – потребителите да могат да оставят и преглеждат отзиви.
  + **Плащания и фактуриране** – интеграция с платежни системи за онлайн плащане и издаване на фактури.
  + **Информационна секция** – важна информация за COVID-19, туристически услуги и актуални оферти.

**2. To Do (за текущия спринт)**

* При всяка спринт планиране среща изберете кои задачи от Backlog-а ще влязат в текущия спринт. Примерни задачи за първия спринт:
  + Създаване на базова структура на приложението (бекенд и фронтенд).
  + Първоначален дизайн на потребителския интерфейс (UI).
  + Базова регистрация и вход за потребители.
  + Създаване на структура за базата данни за хотели и резервации.
* Определете приоритети и конкретен отговорник за всяка задача.

**3. In Progress (В процес)**

* Премествайте задачите тук, докато екипът работи по тях. Добавяйте под-задачи (checklist) за всяка функция, за да е ясно какви стъпки са необходими.
* Пример:
  + Задача: **Създаване на страница за профил на хотел**
    - Под-задачи: дизайн на структурата, добавяне на галерия, интеграция на рейтингова система, визуализация на удобствата.

**4. Testing (Тестване)**

* Когато задачите са готови за тестване, преместете ги в този списък. Например:
  + Тестване на регистрацията и входа с различни сценарии.
  + Проверка на филтрите за търсене по дати и локация.
  + Проверка на страницата за хотелски профили с визуализация на различни елементи.
* Всяка задача за тестване може да има списък с тестови случаи и критерии за приемане.

**5. Done (Готово)**

* Тук премествайте задачите, които са напълно завършени и преминали всички етапи на тестване.
* Например, след като тестовете на страницата за хотелски профили са успешни и всички критерии за приемане са покрити, преместете задачата в „Done“.

**6. Sprint Review**

* Добавете картичка за всеки завършен спринт с описание на основните постигнати резултати и трудности.
* Приемете обратната връзка от клиента и, ако е необходимо, добавете нови задачи в Backlog, които да бъдат адресирани в следващи спринтове.

**7. Sprint Retrospective**

* След края на всеки спринт, създайте картичка с обсъдените точки на ретроспективата: какво е било успешно, какво е предизвикало забавяне, какви подобрения могат да се въведат.
* Примерна картичка:
  + Какво е проработило добре: комуникацията между фронтенд и бекенд екипите.
  + Какво трябва да се подобри: по-бързо идентифициране на блокиращи фактори.
  + Идеи за подобрение: редовна проверка на блокиращите задачи.

**Допълнителни настройки за Trello**

* **Етикети (Labels)**: Създайте етикети за различни категории задачи като „Фронтенд“, „Бекенд“, „Тестване“, „UI/UX дизайн“, „Висок приоритет“ и др.
* **Членове (Members)**: Присвоявайте задачи на конкретни членове на екипа според техните роли.
* **Крайни срокове (Due Date)**: Определяйте срокове за всяка задача, за да се проследява прогресът по спринта.
* **Прикачени файлове и документи**: Можете да добавите дизайн на интерфейса, документ с изисквания или други важни файлове директно в съответната картичка.

**Пример за първи спринт**

**Спринт 1 – Основна структура и начални функционалности (1 седмица):**

* Бекенд: Създаване на базови API за хотели, потребители и резервации.
* Фронтенд: Начална страница и форма за регистрация и вход.
* База данни: Основна структура за хотели и потребителски профили.
* Тестване: Базови функционалности – регистрация, вход и начална страница.

Така структуриран Trello борд ще ви помогне да проследявате всички етапи на разработка и да организирате работата по уеб приложението за морски хотели с максимална ефективност.

Етапите за разработка на уеб приложение за морски хотели с методологията Scrum могат да бъдат разделени на следните основни фази:

**1. Планиране и анализ на изискванията**

* **Събиране на изискванията**: Съберете информация от клиента за функционалностите и целите на приложението. Включва създаване на списък с функционални изисквания (регистрация, резервации, профил на хотел, плащания и т.н.) и нефункционални изисквания (скорост, сигурност, мащабируемост).
* **Формиране на екипа и ролите**: Определете кой ще бъде Product Owner, Scrum Master и кои ще са членовете на екипа (дизайнери, фронтенд и бекенд разработчици, QA инженери).
* **Създаване на backlog**: Превърнете изискванията в потребителски истории, които ще се използват като основни задачи в backlog-а.

**2. Дизайн и прототипиране**

* **Дизайн на потребителския интерфейс (UI)**: Създайте дизайн, който е лесен за навигация и отговаря на нуждите на потребителите (напр. гости, хотелиери).
* **Прототипиране**: Изработете прототип на приложението с основните екрани, за да визуализирате как ще изглежда и функционира интерфейсът. Този прототип може да се сподели с клиента за предварителна обратна връзка.
* **Подготвяне на архитектурата**: Определете основните технологии, които ще се използват (например React или Vue за фронтенд, Node.js или Django за бекенд), и архитектурата на приложението (напр. MVC или микросервизи).

**3. Разработка (с разпределение по спринтове)**

**Първоначална настройка и основна функционалност**:

* **Настройка на разработката**: Инсталирайте всички основни инструменти, които екипът ще използва за разработка, като платформи за контрол на версиите (Git), инструменти за управление на задачи и сървърна инфраструктура.
* **Основни функции и модули**:
  + Първи спринт: създаване на базова регистрация и вход на потребители.
  + Следващи спринтове: добавяне на търсене на хотели, система за резервации, страница за хотелски профили, и т.н.

**Фронтенд и бекенд разработка**:

* **Фронтенд**: Разработване на UI компонентите, базирани на дизайна. Включва функционалности за търсене, преглед на хотели, резервация и управление на потребителския профил.
* **Бекенд**: Разработване на API-та за обработка на заявки, свързани с потребителите, хотелите, резервациите, плащанията и др.

**База данни**:

* Създаване на основните структури за базата данни, включително таблици за хотели, потребители, резервации и ревюта.

**4. Тестване и контрол на качеството (QA)**

* **Unit и интеграционни тестове**: Провеждане на тестове за различни модули на приложението, за да се гарантира, че всяка отделна функция работи правилно.
* **Тестване на потребителски интерфейс**: Проверка на UX/UI компонентите, за да се гарантира, че приложението е лесно за използване и интуитивно.
* **Потребителски и стрес тестове**: Тестове за представяне при натоварване, за да се уверите, че системата ще може да обработва множество заявки едновременно, особено в сезона на натовареност.

**5. Демонстрация и приемане от клиента**

* **Sprint Review**: В края на всеки спринт демонстрирайте на клиента постигнатото, за да получите обратна връзка.
* **Приемни тестове**: Извършете приемни тестове с участието на клиента, за да се гарантира, че всички функционалности отговарят на изискванията.
* **Корекции и подобрения**: Въз основа на обратната връзка добавете нови задачи в backlog или направете корекции в съществуващите.

**6. Деплоймънт и внедряване**

* **Подготовка на сървърна среда**: Настройте сървърите и инфраструктурата за приложението.
* **Деплоймънт в производствена среда**: Публикувайте приложението на съответния сървър и се уверете, че всичко работи коректно.
* **Мониторинг**: Настройте системи за мониторинг, които ще проследяват производителността и ще предупреждават за евентуални проблеми.

**7. Поддръжка и надграждане**

* **Оперативна поддръжка**: Подсигурете екип за бърза реакция при евентуални проблеми и грешки.
* **Надграждане**: След стартирането на приложението събирайте обратна връзка от реални потребители и клиенти, като добавяте нови функционалности и подобрения в backlog за бъдещи версии.

Тези етапи ще ви осигурят систематичен подход за изграждането на уеб приложението, като методологията Scrum ще позволи гъвкавост в адаптацията на изискванията и ефективно изпълнение на всяка фаза.