**На примере информационной системы для управления версиями кода - Git:**

1. **Определить предметную область**:

Предметная область - это управление версиями кода.

Это включает в себя отслеживание изменений в коде, сохранение различных версий кода, слияние изменений от разных разработчиков, управление ветками кода и т.д.

1. **Провести анализ предметной области**:

Анализ предметной области может включать в себя изучение существующих систем управления версиями, таких как SVN или Mercurial, а также анализ потребностей пользователей, которые будут использовать систему.

Это может включать в себя опросы, интервью и наблюдение за работой разработчиков.

Например, какие функции они часто используют, какие проблемы они сталкиваются при использовании текущих систем и т.д.

1. **Проанализировать осуществимость**:

Осуществимость проекта может быть оценена по нескольким критериям, включая техническую осуществимость (есть ли необходимые технологии и навыки), экономическую осуществимость (будет ли проект рентабельным) и юридическую осуществимость (соответствует ли проект законодательству).

Например, нужно будет убедиться, что есть разработчики с навыками работы с Git, что проект будет приносить достаточно дохода для окупаемости и что он не нарушает никаких законов о защите данных или интеллектуальной собственности.

1. **Формирование и документирование требований к проекту**:

Требования к проекту могут включать в себя функциональные требования (что система должна делать), нефункциональные требования (такие как производительность, безопасность и надежность) и требования к пользовательскому интерфейсу.

Эти требования обычно документируются в виде спецификации требований к программному обеспечению.

Например, функциональные требования могут включать в себя возможность создавать новые репозитории, делать коммиты, создавать ветки и т.д.

Нефункциональные требования могут включать в себя требования к производительности (например, сколько коммитов система должна обрабатывать в секунду) и требования к безопасности (например, как система должна хранить и передавать данные).

1. **Подготовить примерное техническое задание проекта**:

Техническое задание может включать в себя описание архитектуры системы, выбор технологий, план разработки и тестирования, а также оценку затрат и сроков.

Например, можно выбрать использовать Python для разработки серверной части системы и React для разработки клиентской части.

Можно планировать выпускать новые версии каждые две недели и использовать автоматизированное тестирование для обеспечения качества кода.

А также должны оценить, сколько времени и денег потребуется на разработку, тестирование и поддержку системы.