1. Feature driven development

**Step 1: Develop the overall model**

This step is where you write your outline to define your domain model—the business problem you want your software development project to solve. The team works closely with the chief architect to define the scope and context of the system. Multiple domain models should be merged into one overall model as an outline for your system.

**Step 2: Build a features list**

The features list is similar to the scrum product backlog. Identify the features that are important to the client. Features are expressed as action, result, and object (for example, “validate the account number of the user”).

It should not take longer than two weeks to develop any given feature. If a feature will take longer than two weeks to build, it should be broken up into smaller features.

**3: Plan by feature**

Determine the order in which the features from your features list will be developed and implemented. Consider potential risks, dependencies, team and individual workload, and any other obstacle that could hinder feature development.

Then assign feature sets to the programmers who are most capable and have the bandwidth to develop them within the specified timeframe.

**4: Design by feature**

The chief programmer determines which features will be designed and built in a two-week iteration. This person also defines the feature priorities and determines who will be involved on the feature team. A design review needs to be completed by the whole team before moving forward.

**5: Build by feature**

In this step, all of the items necessary to support the feature design are implemented. The user interface is designed, and a feature prototype is built and tested. If the feature passes the test and is approved, the completed version can be added to the main build and made available to customers.

Overall model:  
build a calendar app on the platform of Teams to associate users to schedule meetings with others.

Feature list:

1. show user’s calendar on the app of Teams.

2. import/export calendar files in csv.

3. create new calendar and add events on calendar.

4. invite others to meetings (UI-tab, email, reminder).

5. users select the preference and system automatically schedule meetings.

6. take location and transportation into consideration to decide time slot.

Plan by feature:

Order

First know about the Team app platform. Using Visual Studio in C#. How to implement it. Some good examples.

2. IOS app

Calendar:

a mix of Outlook and Google calendar account to see calendars.

Check events on calendar.

Add events.

Task and reminder.

Export calendar in pdf.

No meeting scheduling function.

3. Microsoft Teams platform

I am following the guide (<https://docs.microsoft.com/en-us/microsoftteams/platform/tutorials/get-started-dotnet-app-studio>)

Meeting errors when building the project in Mac.