Waterfall Model:

Software Specification:
Feasibility – This project is possible with the current technology available to us, including databases and user interfaces. The cost of this project is also within budget.

Requirements and Specifications:

Software is able to support a different variety of users

- All users will be able to see and change their personal information and view schedules
    - Display weekly class schedule for user
    - Display personal information for user in table
- Students will be able to add and drop classes, requestion meetings with advisors, review their degree audit and required classes
    - Students will be able to see major, minor, and personal informations when requested
    - Able to set up and adjust meetings with advisors and other faculty
    - Able to add and drop classes from their schedule
- Professors will be able to view their class information and classlist, review and update job specific information and ...
    - Professors can view and update their class information
    - View a classlist for all of the classes they are teaching
    - Review and update their job specific information
- Admins will be able to update student and teacher databases, access to testing accounts, and implements new updates to the system

Design:
Users – Each type of user will have their own class with their respective parameters and functions. All users will have access to global functions

Calendar – The software will have a calendar where users will be able to view their weekly class schedule and any meetings they have scheduled for that week.

User Interface – The software will have a UI where the users will be able to navigate different tabs to get different information or update schedules, and

Database Design – The database will be storing all the users information including id, name, and user specific information like major or department.

Validation – The software will have testing accounts available to the admins so they will be able to test all functionality and set up automated test with the accounts to make sure there are no bugs.

Incremental

1. Users should be able to request information regarding their schedules and personal information that comes from a SQL database.

2. Users should be able to request information through a UI, and schedule meetings with advisors and professors should be able to update their office hours and basic test should be implemented to reduce bugs.
3. Users should be able to use UI to add/drop and view their class schedules, updates meetings, and access the calendar. Testing should cover the whole software product and be automated using a testing framework.

Integrate and Configure

1. Use Visual Studio and C# to develop classes and functions that will be able to store user information and show information to the user through the command line.
2. Use PostgreSQL and a linux host to host a database to store all users information and link it to visual studio through ... package
3. Use Qt to develop a user interface for the software instead of using the command line. Underneth the UI, the code we developed before implementing the UI will be able take in user inputs and display and update what the user needs.
4. Use Nunit and Selenium to develop tests for our software by simulating button clicks and keyboard strokes that would be made by a user.