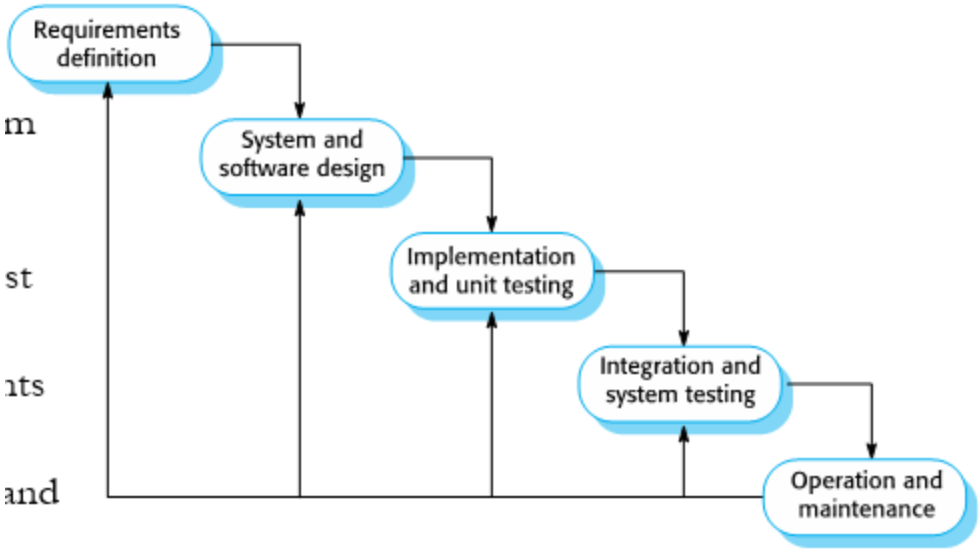

Requirements definition:

1. Feasible study
    a. Yes we need classes and objects, a database, and a user interface
2. Requirements elicitation
    a. We ask what kind of the schedule the school wants in terms of information, survey the students how they would want the process to go.
3. Requirement specification
    a. Student users should be able to search for courses, add/drop courses, and print their schedules
    b. Teacher users should be able to print their schedules, print their class list, and search for courses
    c. Adim users should be able to add/remove courses from the system, add/remove users, add/remove students from courses, search and print roasters and courses
4. Requirements validation
    a. The group would check the other specific requirements.

Design and implementation

1. Architecture design
    a. Some components would be classes and object, function, database, and GUI
2. Interface design
    a. You would be able to in the code they were able to call back to the main file
3. Component design
    a. The user inface will be text based where you would be able to input commands to use it.
4. Data base design
    a. User: first name, last name, ID
    b. Student : first name, last name, ID, title, courses

    c. Teacher: first name, last name, ID, title courses, roaster
    d. Admin: first name, last name, ID, title courses, roaster, users
5. Software validation
    a. Components testing
        i. Testing the code seeing the users are able to do what they can do.
    b. System testing
        i. Test the system after integrating
    c. Acceptance testing
        i. Present it to school and let them test it out with “real” students, teachers, and admin
6. Software evolution
    a. After the years we could modify it such as an easier interface or if the users have suggestions over the years we could implement it.