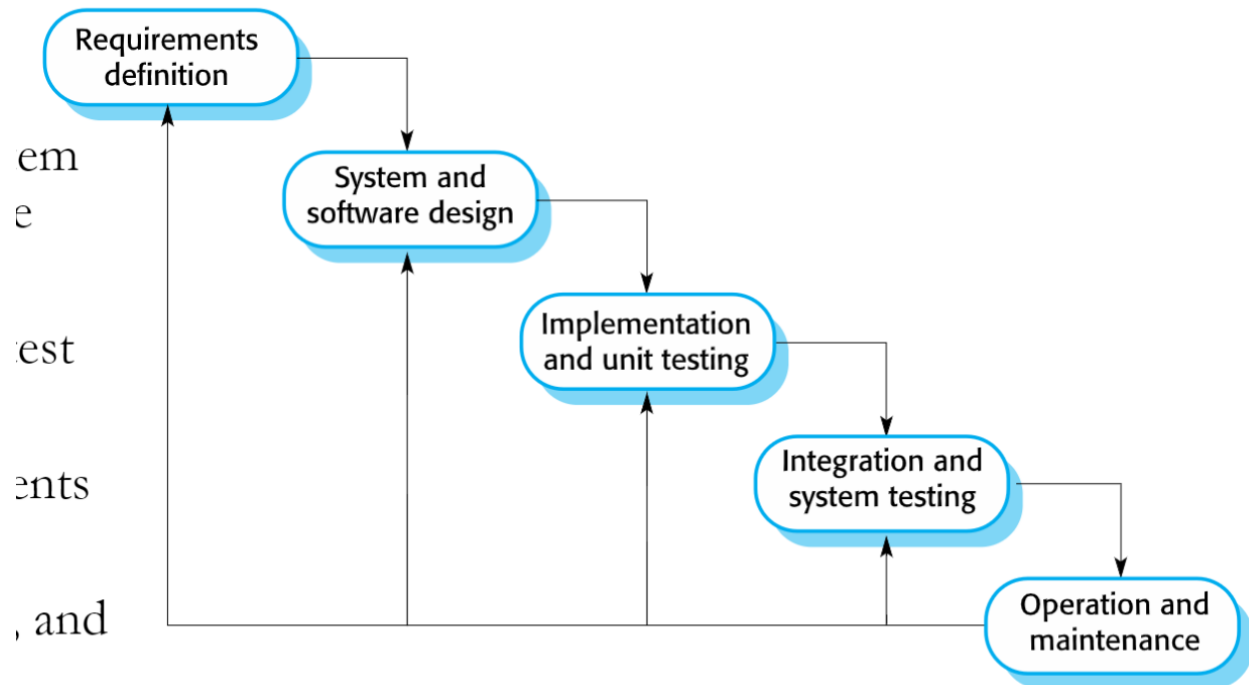


Waterfall



Requirement analysis + definition

Goals:

- Design a university scheduling system (similar to LeopardWeb) that allows students, faculty, and admin to add courses, search for courses, print schedules, etc.
- System should include multiple semesters, print-out of schedule, and scheduling preferences

Constraints:

- **Students** can only search courses, add/drop courses, and print their schedule
- **Instructors** can only print their schedule, print their class list, and search for courses
- **Admins** can only add/remove courses to the system, add/remove users, add/remove students from a course, search/print rosters and courses
- System should work for 100 students, 10 instructors, and 1 admin

Services:

- Databases for students, admins, instructors, CRN's, course names, scheduling times, and rosters

System software and design

1. Display login page for username + password.
2. Prompt user to choose if they are:
 - a. Student

- b. Instructor
 - c. Admin
- 3-1. If the user is a Student, prompt the user to either:
 - a. Search a course
 - b. Add/drop a course
 - c. Print their schedule
- 3-2. If the user is an Instructor, prompt the user to either:
 - a. Print their schedule
 - b. Print their class list
 - c. Search for courses
- 3-3. If the user is an Admin, prompt the user to either:
 - a. Add/remove courses from the system
 - b. Add/remove users from the system
 - c. Add/remove students from a course
 - d. Search/print roster
 - e. Search/print courses

Implementation and unit testing:

- Base class "User" created with all three subclasses (Student, Admin, and Instructor)
- Create databases for students, admins, instructors, CRN's, course names, scheduling times, and rosters

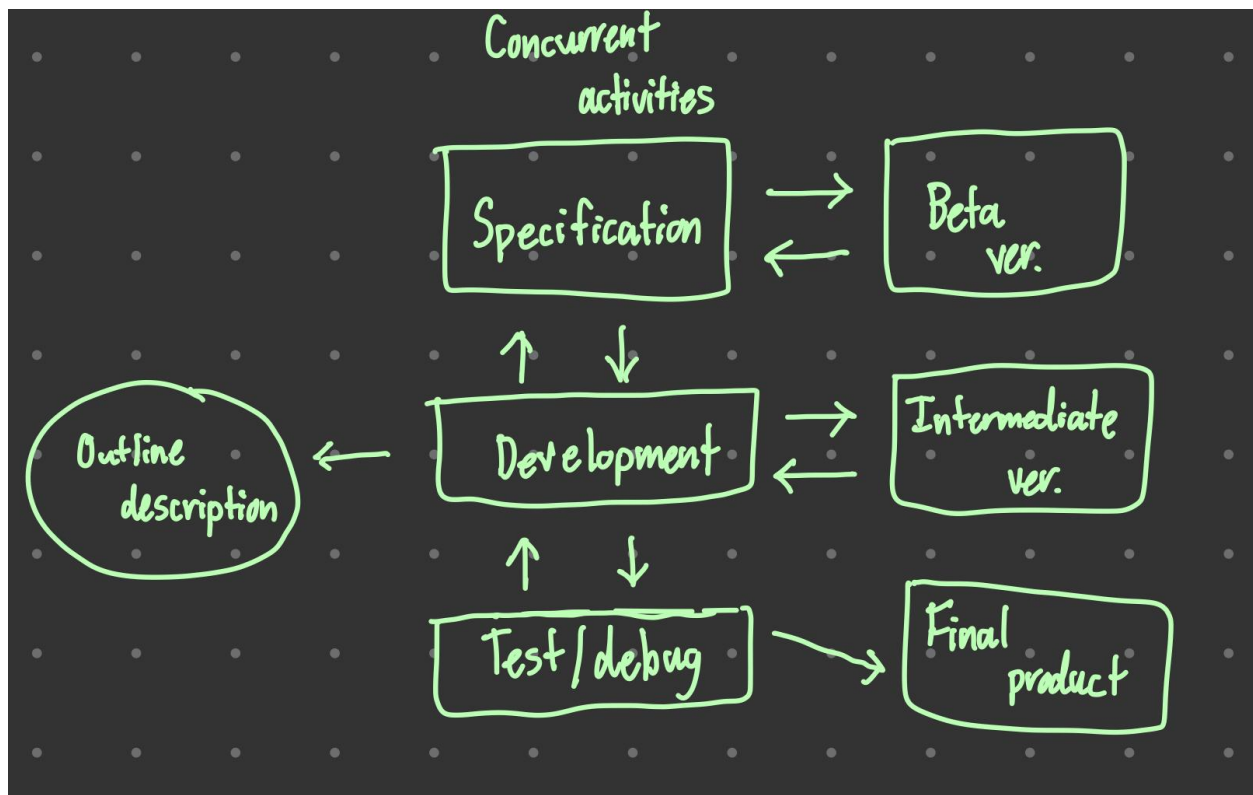
Integration and system testing

- Main program constructed with called classes, login capability, and system navigation

Operation and maintenance:

- System program compiled and tested:
 - 1. Any user has the ability to navigate the system
 - 2. Any bugs/issues will be fixed throughout the program

Incremental



Outline description:

- Create a scheduling system (similar to LeopardWeb) that allows students, faculty, and admin to add courses, search for courses, print schedules, etc.
- Base class: User
- Sub-classes: Student, Instructor, Admin
- Students can:
 1. Can search courses
 2. Add/drop courses
 3. Print their schedule
- Instructors can:
 1. Print their schedule
 2. Print their classlist
 3. Search for courses
- Admins can:
 1. Add/remove courses from the system
 2. Add/remove users from the system
 3. Add/remove students from the course
 4. Search/print rosters and class lists

Development:

- Classes are created with their corresponding methods, constructors, and attributes

Specification:

- Student, instructor, and admin classes are created

- Attributes: First name, last name, ID

Test/Debug:

- Run multiple test cases for students, instructors, and admins
- Troubleshoot any errors within the program

Beta ver:

- First draft at functional scheduling system

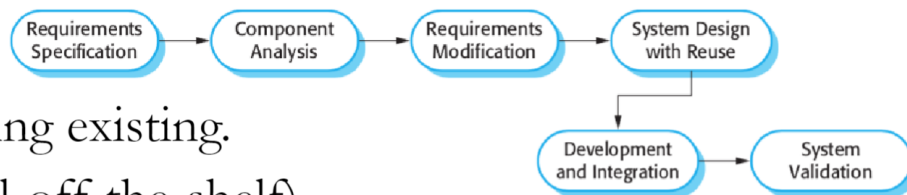
Intermediate ver:

- Improved code with a combination of major and minor changes

Final product:

- Completed scheduling system (similar to LeopardWeb) that allows students, faculty, and admin to add courses, search for courses, print schedules, etc.

Integration and configuration:



ing existing.

al-off-the-shelf)

Requirements specifications:

- Create a scheduling system (similar to LeopardWeb) that allows students, faculty, and admin to add courses, search for courses, print schedules, etc.
- Base class: User
- Sub-classes: Student, Instructor, Admin
- Students can:
 1. Can search courses
 2. Add/drop courses
 3. Print their schedule
- Instructors can:
 1. Print their schedule
 2. Print their classlist
 3. Search for courses
- Admins can:
 1. Add/remove courses from the system
 2. Add/remove users from the system
 3. Add/remove students from the course
 4. Search/print rosters and class lists

Requirement modification:

- Base class "User" created along with subclasses Student, Instructor, and Admin

Component analysis:

- Subclasses created to manipulate the User class

System design and reuse:

- Subclasses are used to manipulate these databases:
- 1. Class list
- 2. Schedule
- 3. Student list

Development and integration

- Test the program using different test cases
- Beta code development (after feedback)
- Bug fixes

System validation

- Troubleshoot bugs within system
- Final product ready for deployment