Assignment 2  
Daniel Khoury  
ELEC 3225  
May 19th, 2024

**Waterfall Model:**

The Waterfall Model involves a linear and sequential approach to software development. Here is a detailed process model diagram and description using the Waterfall method:

1. Requirements Analysis (2 weeks)

* Gather all functional and non-functional requirements.
* Define detailed requirements for each type of user (student, instructor, admin).
* Specify database requirements for users and courses.
* System Design (3 weeks)

1. Design the overall system architecture.

* Create detailed design documents for the database schema, user interfaces, and backend logic.
* Design the base class User and derived classes Student, Instructor, and Admin.
* Implementation (4 weeks)

1. Develop the database schema and set up the database.

* Implement the base class User and derived classes.
* Develop functionalities for students (search courses, add/drop courses, print schedule).
* Develop functionalities for instructors (print schedule, print class list, search courses).
* Develop functionalities for admins (add/remove courses, add/remove users, manage schedules).
* Integration and Testing (3 weeks)

1. Integrate all components of the system.

* Perform unit testing for individual components.
* Conduct system testing to ensure the entire system works as expected.
* Deployment (1 week)

1. Deploy the system to the university's servers.

* Conduct initial user training for students, instructors, and admin.
* Maintenance (Ongoing)

1. Monitor the system for bugs and issues.

* Provide ongoing support and updates as needed.

Proposed Timeline:

* Requirements Analysis: Week 1-2
* System Design: Week 3-5
* Implementation: Week 6-9
* Integration and Testing: Week 10-12
* Deployment: Week 13
* Maintenance: Ongoing

**Incremental Development:**

Incremental Development involves building the system in small, manageable increments, adding functionality in each iteration. Here’s a proposed process model:

Iteration 1: Core System (4 weeks)

* Implement the base class User.
* Develop the database schema.
* Implement basic functionalities: user registration, login, and profile management.
* Develop basic student functionalities: view available courses, print schedule.

Iteration 2: Expanded Student and Instructor Features (4 weeks)

* Complete student functionalities: add/drop courses.
* Implement instructor functionalities: view schedule, view class list, search courses.

Iteration 3: Admin Features (4 weeks)

* Implement admin functionalities: add/remove courses, add/remove users.
* Implement admin functionalities to manage student registrations.

Iteration 4: Enhanced Features and Testing (4 weeks)

* Add advanced features: multiple semesters support, scheduling preferences.
* Conduct thorough testing for all functionalities.
* Integrate and test all components together.

Proposed Timeline:

* Iteration 1: Week 1-4
* Iteration 2: Week 5-8
* Iteration 3: Week 9-12
* Iteration 4: Week 13-16

**Integrate and Configure:**

The Integrate and Configure method involves using existing software components and integrating them to build the system. Here’s a detailed approach:

1. Identify Components (2 weeks)

* Search for existing open-source or commercial databases suitable for the system
* Look for user interface frameworks (e.g., Bootstrap, AngularJS).
* Find scheduling and calendar libraries (e.g., FullCalendar for scheduling).

1. Configure Databases and Backend (3 weeks)

* Set up and configure the chosen database.
* Implement the base class User and derived classes using an existing ORM (Object-Relational Mapping) framework

1. Integrate User Interface Components (3 weeks)

* Develop the user interfaces using chosen frameworks.
* Integrate frontend with backend using RESTful APIs.

1. Add and Configure Functionalities (4 weeks)

* Configure and integrate functionalities for students, instructors, and admins.
* Utilize existing libraries for specific tasks (e.g., FullCalendar for scheduling).

1. Testing and Refinement (3 weeks)

* Perform integration testing to ensure all components work together.
* Conduct user acceptance testing to gather feedback and make necessary adjustments.

1. Deployment and Training (1 week)

* Deploy the system on the university’s servers.
* Provide training sessions for all user types.

Proposed Timeline:

1. Identify Components: Week 1-2
2. Configure Databases and Backend: Week 3-5
3. Integrate User Interface Components: Week 6-8
4. Add and Configure Functionalities: Week 9-12
5. Testing and Refinement: Week 13-15
6. Deployment and Training: Week 16

Sources and Components:

1. Database: MySQL or PostgreSQL (widely used, reliable).
2. User Interface: Bootstrap (responsive design), AngularJS (dynamic web apps).
3. Scheduling: FullCalendar (versatile calendar library).

By following these models, the scheduling system for the university can be developed effectively, ensuring all requirements are met and the system functions efficiently for all user types.