

Nick Cebula

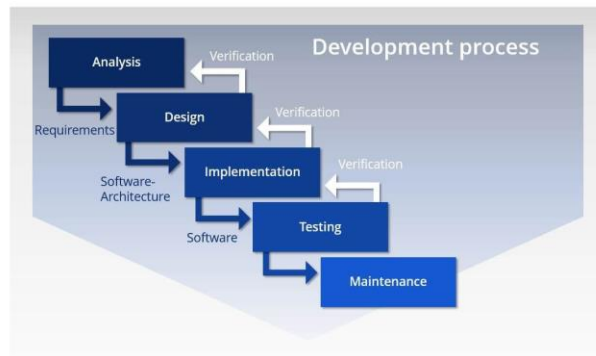
ELEC 3225 APC

Professor Dow

May 21, 2024

Assignment 2 Part 1

a. Waterfall diagram



- b. The waterfall model is a linear approach to software development, the result of every phase should be checked back for verification with other steps. You should not start until the previous phase is finished.

You start off with understanding what the project aims to achieve and gather further information and requirements for the project. Once you satisfy the requirements you can move onto developing a design that will meet the requirements.

This could include a high-level program design, flow charts, and specify hardware and software components.

Once you reach the software architecture and verify that it meets requirements, you can translate the system design into code of individual modules where you will have source code and test each of the individual modules. Before moving onto testing, you must go back and verify that it fits the design.

Then you will compile the modules into a complete software program. Then you will do a system test, perform bug fixes, check for performance and security. You must go back to the implementation of the software phase for verification that it meets previous requirements. All of the phases are interconnected.

After this, you will be able to deploy the program to users. You will ensure that the system continues to operate efficiently after deployment. This includes monitoring, continuing bug fixes and pushing out system updates to maintain an effective system based on user feedback.

- c. For the University schedule system project, here's a plan on how you can apply the waterfall model.

You would start off with understanding the constraints and goals in the project description. You will understand how each of the classes connects to each other, and how they are based off the User class and the information each must contain in its attributes and methods. You also need to develop a game plan on how the schedule system will work for 100 students, 10 instructors, and 1 admin. Aswell, as how you can meet the requirement of including multiple semesters and preferences.

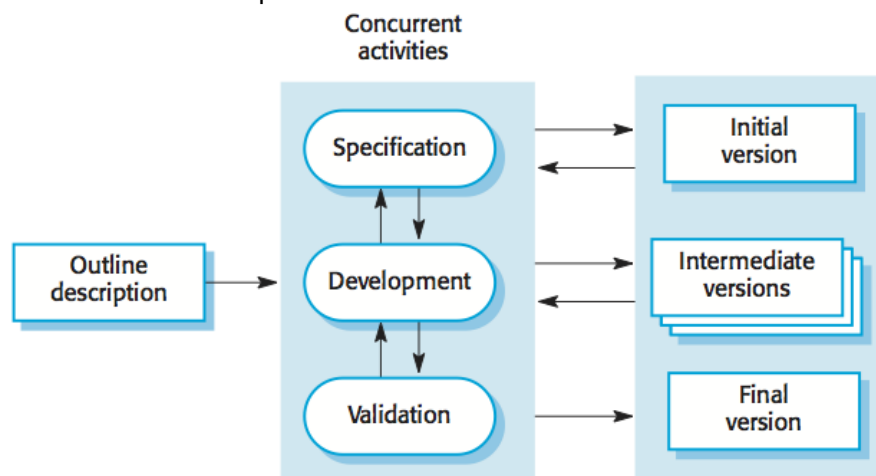
After you know how you can approach a program to meet the requirements, you can start with system software and design. Keeping the requirements in mind, you should start writing flowcharts/ plans for all the classes for each of the users first. Plan the user interface and interaction flow for different user roles. Then write how you will define methods for course registration, scheduling generation, and searching classes based on your class type.

Check back with the requirements before moving onto implementation. Now, you can write the code in your preferred IDE and language. Making classes and derived classes. Next you will implement the database for the adequate user size. Develop methods for each of the classes while connected to the database, based on user permission. Create an interface and make sure it connects to the backend (database) correctly. Integrate semester management and schedule features.

Check back with the previous steps to make sure that the tested program works. After this you can integrate all the files into one program and begin user testing of each of the methods for each of the classes. Ensure that the system can handle all of the active users at once.

After testing functionality, speed, and fixing bug fixes, continue operating on bugs and monitor feedback after deployment. Update the system based on any new requirements.

2. Incremental Development Model



a.

- b. To start this process you should outline the description by defining the overall program objective and major components that need to be developed. Plan the increments in order that they should be developed. Then develop the initial version, allow the user to test it. Then develop other intermediate versions, increase the functionality, and validate them until you have a final version developed.
- c. i. The first subsystem to be developed first is the user class, you would test it in the main making sure all the methods and attributes work.
- C ii. The second version would include the admin, instructor, and student classes with all of their unique methods such as add/drop classes and printing schedule for student/instructor. And the classes with higher classes would have more control over the schedule such as removing/adding students. You would test this in the main validating inheritance and functionality.
- c.iii. The final system would include the database where the methods would refer to the database with the list of classes and students. The functionality will be tested in the main code allowing each type of user to perform their unique methods properly.

3. Integration and Configuration model

- a. For this model it is focusing on reusing existing code and validating steps along the way with using previous progress. You start off by analyzing the initial components. The intermediate stages would find and study exiting software while modifying original code along the way. As well as modifying the requirements as you go to fine tune to the goal. The programmer will keep developing and integrating until they reach a final version that can be validated to the system requirements.
- b. For the university scheduling system, a library that could be used is SQLAlchemy, which is an object relation database library. It would be very helpful because you can make classes for each type of user and then populate them. You can use this by importing the library. It would use lists for each variable depending on the type of user.

Another option is Django which is an open-source python based web framework for working with databases.

We could also use Pandas which is very good at managing large amount of data and preforming any calculations, which could be based off a excel format.

I would also use tkinter for the interface, I have used it before and it is user friendly and easy to program, that way we can see all the information that we need to, and printing schedule and current classes would be appealing to read. You can configure this by adding all the text boxes for each user and then a side bar which prints the information you need to see.

Example: student will be able to have name, grade, ID, maor, and schedule. Then to meet the specification you can have 100 instances of student.

```

engine = db.create_engine('sqlite:///datacamp.sqlite')
conn = engine.connect()
metadata = db.MetaData()

Student = db.Table('Student', metadata,
                    db.Column('Id', db.Integer(), primary_key=True),
                    db.Column('Name', db.String(255), nullable=False),
                    db.Column('Major', db.String(255), default="Math"),
                    db.Column('Pass', db.Boolean(), default=True)
                    )

metadata.create_all(engine)

```

Figure 1: Code snippet for DataCamp showing integration of student class in the database.

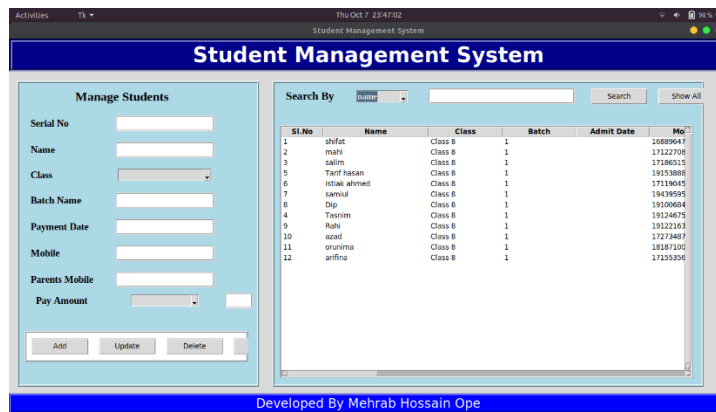


Figure 2: Existing school management system example using Tkinter

References: <https://pypi.org/project/SQLAlchemy/>
<https://www.datacamp.com/tutorial/sqlalchemy-tutorial-examples>
<https://images.app.goo.gl/b1rsHAGVBHBxasMs8>
<https://github.com/QasimInayat/School-Management-System-Python---Tkinter-/blob/master/Home.py> (code for existing school management system in Python using tkinter for interface)

Using the waterfall model as a backbone would be beneficial because each of the stages must be approved to the overall requirement definition before moving onto the next stage. It is also great to have other stages overlapping to feed information to each other. But I think we should be able to move onto future stages while fine tuning other stages. So, I would use the incremental model after having the outline done with the waterfall model. This is because with my programming experience it is helpful to move on so I have a better understanding of how other modules can be improved upon while not spending too much time on one step. Incrementing helps add overall functionality while being able to get feedback on previous methods. It is also time friendly since we only have so much time to complete this project throughout the semester. This would allow us to research other user codes and how they perform along with other libraries we can implement.