# CS 255 Model Application Short Paper

Tyler Daniels

Tyler.daniels1@snhu.edu

Southern New Hampshire University

## Process Model Application

The DriverPass scenario revolves around the development of a digital platform aimed at facilitating the issuance and management of driver's licenses. This scenario presents a complex problem that requires a systematic approach to design and implementation. To effectively address this challenge, applying a process model is essential. In this paper, we will explore the application of a process model to the DriverPass scenario, with a focus on the stages and steps involved in the design and development process. Specifically, we will utilize the Systems Development Life Cycle (SDLC) as the chosen process model due to its well-structured approach to managing complex projects.

The Systems Development Life Cycle (SDLC) is a structured approach to project management and software development that divides the entire process into distinct phases. These phases provide a roadmap for planning, designing, building, testing, and deploying a system. The SDLC ensures that each phase is completed thoroughly before proceeding to the next, minimizing errors and enhancing the overall quality of the final product. The key phases of the SDLC include planning, analysis, design, implementation, testing, deployment, and maintenance.

Applying the Systems Development Life Cycle (SDLC) to the DriverPass scenario offers a systematic and structured approach to designing and implementing the digital platform for driver's license issuance and management. By following the phases of the SDLC, the project team can ensure that the platform is developed efficiently, meets stakeholder requirements, and is sustainable in the long term. Proper planning, analysis, design, implementation, testing, deployment, and maintenance are all essential steps in successfully bringing the DriverPass scenario to fruition. This approach not only enhances the quality of the final product but also minimizes risks and maximizes the chances of success in a complex project like DriverPass.

## Object Model Application

To apply an object model to the design for the DriverPass scenario, we can identify key objects, their attributes, and relationships within the system. Here's a simplified representation of how an object model might look for this scenario:

1. User Object:

Attributes: First Name, Last Name, Address, Phone Number, State, Username, Password (hashed and salted), Credit Card Number, Expiration Date, Security Code.

Relationships:

A User can have multiple Driving Appointments.

A User can have multiple Driving Lessons.

A User can have a Role (e.g., Administrator, Secretary, Student).

1. Driving Appointment Object:

Attributes: Appointment ID, Date, Time, Pickup Location, Drop-off Location.

Relationships:

Belongs to a User.

Is scheduled with a Driver.

Can belong to a Package (e.g., Package One, Package Two, Package Three).

1. Driver Object:

Attributes: Driver ID, Name, Assigned Car.

Relationships:

Can have multiple Driving Appointments.

Can have multiple Driving Lessons.

1. Package Object:

Attributes: Package ID, Description, Duration (in hours).

Relationships:

Contains multiple Driving Lessons.

Can be associated with multiple Driving Appointments.

1. Driving Lesson Object:

Attributes: Lesson ID, Duration (in hours), Lesson Type (e.g., In-Car, In-Person, Online).

Relationships:

Belongs to a Package.

Can be associated with multiple Driving Appointments.

This object model provides a foundation for representing the key elements and relationships within the DriverPass system. It allows for a structured approach to designing and implementing the system, making it easier to understand, maintain, and extend in the future. Depending on the complexity of the system and specific requirements, this model can be further refined and expanded.

## Process and Object Model Comparison

In the context of the DriverPass scenario, there are several project management models or methodologies to consider, each with its own set of advantages and disadvantages.

The Waterfall Model is advantageous for projects with clear and stable requirements, making it suitable for well-defined scopes. It allows for structured and phased development and emphasizes documentation, which can be crucial for compliance. However, it lacks flexibility, collects user feedback late in the project, and can result in longer delivery times.

The Agile Model provides flexibility and is adaptive to changing requirements, making it suitable for industries like driving education where regulations and customer preferences change frequently. It encourages early and continuous user feedback and supports iterative development. However, it can introduce complexity in large projects, prioritize working software over documentation, and potentially lead to scope creep.

A Hybrid Model, combining Waterfall and Agile elements, offers a tailored approach. It allows for a balance between flexibility and structure, making it advantageous for projects with a mix of stable and evolving requirements. However, managing a hybrid model can be complex and may lead to conflicts between the two approaches.

The DevOps Model supports continuous integration and delivery, aligning well with the need for frequent updates and compliance with DMV regulations. Automation reduces errors and increases efficiency, but implementing DevOps practices may require a cultural shift and significant resource investment.

The Incremental Model divides projects into manageable stages, useful for complex systems. It allows for early deliveries of usable products but requires careful dependency management and may not suit projects with well-defined and unchanging requirements.

Ultimately, the choice of the project management model for the DriverPass scenario should consider factors such as project size, complexity, regulatory requirements, customer feedback expectations, and organizational culture. A hybrid approach that combines Waterfall and Agile elements may offer a practical solution to balance regulatory compliance and agility.