# CS 255 Model Application Short Paper

Bryan Howard

Bryan.howard2@snhu.edu

Southern New Hampshire University

## Process Model Application

[How would you apply a process model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.]

Applying an Iterative Process Model or Agile methodology to the design of the DriverPass system would be highly effective given the evolving nature of the project. This approach allows flexibility to adapt to changing requirements. The process begins with requirement gathering and analysis, where features like scheduling lessons, tracking customer progress, ensuring role-based access, and maintaining security are collected from the client. Detailed functional specifications are created to outline what the system needs to accomplish, but as the project evolves, new requirements may emerge, making the iterative model particularly useful for incorporating feedback and refinement.

In the design phase, the system architecture and interface are defined, focusing on a responsive web-based design that works across devices. The back-end structure is developed to manage users, lessons, and scheduling. Instead of tackling the entire system at once, the iterative model encourages breaking down the design into smaller, manageable parts. The first designs focus on fundamental features like user registration, lesson scheduling, and progress tracking, while more complex elements such as reporting tools and security layers are tackled in subsequent iterations.

During the development phase, core features are built incrementally. For DriverPass, this starts with user registration and lesson booking. These functionalities are developed, tested, and integrated into the system before moving on to administrative controls, user tracking, and integration with DMV updates. After each cycle, testing and validation are conducted to ensure the new features work properly, preventing disruptions and catching issues early. This ensures a high-quality system, tested after each iteration, before moving on to more complex features.

Finally, deployment and maintenance are managed incrementally as well. The first deployment may include basic functionality like lesson scheduling, while advanced features like payment gateways and reporting tools are rolled out later. Maintenance ensures that the system evolves with the business, addressing new requirements or fixes in future iterations. The Iterative Process Model is ideal for DriverPass because it allows flexibility, early deployment of core features, and continuous improvement, ensuring the system meets the client’s evolving needs while reducing risks at each step.

## Object Model Application

[How would you apply an object model to a design for the DriverPass scenario? Remember, you do **not** need to create diagrams for this paper.]

To apply an object model to the DriverPass system, the key step is identifying the main objects or entities involved, defining their attributes and behaviors, and establishing relationships between them. In this system, core objects include User, Customer, Admin, Lesson, Car, Driver, Progress, and Training Package. Each object has specific attributes and methods. For example, a Customer can schedule, modify, or cancel lessons, while an Admin can manage user accounts, enable/disable training packages, and track system activities. The Lesson object links Customer, Driver, and Car, holding information about lesson details such as time, date, and status.

The relationships between these objects are crucial. A Customer is a type of User, inheriting basic attributes and behaviors while adding functions like booking lessons. Similarly, Lessons connect Customers to Drivers and Cars, ensuring scheduling and management. Object-oriented principles such as inheritance and polymorphism help make the system modular and flexible. Inheritance allows classes like Customer and Admin to share common functionality from the User class, while polymorphism ensures that different users interact with the system in ways appropriate to their roles.

Encapsulation plays a key role in securing data, such as passwords, by restricting direct access and providing secure methods for managing sensitive information. This object model provides a clear, scalable structure for the DriverPass system, supporting modularity, security, and ease of future expansion.

## Process and Object Model Comparison

[What are the advantages of each model for the DriverPass scenario? What are the disadvantages of each model for the DriverPass scenario?]

The Iterative Process Model and Agile Methodology both offer several advantages for the DriverPass system. One key advantage of the Iterative Process Model is its flexibility, allowing continuous feedback and adaptation as new requirements emerge during development. This is important for DriverPass, where features like scheduling, tracking progress, and DMV compliance may evolve over time. Additionally, the model supports early delivery of core features, meaning essential functions like lesson booking can be deployed early, enabling DriverPass to start operations while more complex functionalities are developed in later iterations. Another benefit is reduced risk, as issues can be identified and resolved early during testing phases. Finally, the iterative model allows for incremental improvements, ensuring the system evolves gradually in response to real-world use and feedback.

However, the Iterative Process Model also has some disadvantages. One challenge is the complexity in managing multiple iterations, which requires careful planning and resource allocation to ensure all parts of the system are developed on time. There's also a risk of scope creep, where additional features are introduced during development, potentially leading to delays and increased costs. Additionally, because the system is developed in increments, the early versions may feel incomplete, with key functionalities missing in the initial stages of deployment.

On the other hand, the Agile Methodology provides benefits such as high client involvement, encouraging close collaboration between the development team and DriverPass stakeholders. This ensures that the system aligns closely with client needs and allows for quick adjustments based on feedback or changes in requirements. Agile’s sprint-based approach makes it highly adaptive to change, which is ideal if DriverPass encounters evolving customer demands or new regulatory requirements. However, Agile also has its challenges. It requires constant client participation, which may be difficult for DriverPass if stakeholders cannot commit to regular meetings and reviews. There is also a risk of unfinished features if development tasks are not completed within the sprint timeframe. Additionally, Agile tends to emphasize working software over documentation, which may result in insufficient documentation that could cause issues for future developers or system maintainers.

Overall, both models offer flexibility and adaptability, which are crucial for the DriverPass project. However, they also introduce challenges related to managing scope, client involvement, and documentation. The choice between the two will depend on how DriverPass manages these trade-offs and prioritizes its development needs.

## References

[You were **not** requiredto use external resources for this report. If you did not use any resources, you should remove this entire section. However, if you did use any resources, including the textbook, to help you with your interpretation, you **must** cite them. Use proper APA format for citations. Two samples have been included below for you to use as a reference.

For a book:

Author, A. A. (Year of publication). *Title of work: Capital letter also for subtitle*. Location: Publisher.

For a website:

Author, A. A. (Year, Month Day). Title of article. *Title of Website*. Retrieved from  
https://www.homeaddress.com/