# CS 255 Model Application Short Paper

LLoyd Askew

lloyd.askew@snhu.edu

Southern New Hampshire University

## Process Model Application

The Unified Process (UP) model would be a suitable choice to design the DriverPass scenario utilizing a process model. Throughout all phases of the software development life cycle, the UP frameworks, which is renowned for its iterative and incremental character, lays a heavy focus on flexibility and adaptability.

The UP would first include an inception phase where the project's scope, purpose, and objectives are determined. The project team would undertake a first interview and requirements analysis at this phase, integrating with the data gleaned from the client meeting, and engaging stakeholders from DriverPass. The elaboration stage would come after the inception stage. An in-depth examination of the requirements, system architecture, and creation of a solid project plan would be part of this stage. According to the DriverPass scenario, this stage would involve having a complete grasp of their unique requirements, from the online testing platform to the availability of on-road training options.

The construction phase, the subsequent stage, would be focused on system implementation. It would encompass the actual development of the online platform, integration of features, and ensuring that the system aligns with the specified requirements. This phase would be critical for the creation of a functioning system that caters to the diverse needs of DriverPass.

Lastly, the transition phase would involve product testing, deployment, and training activities. Rigorous testing of the system's functionalities, its integration with databases, and end-to-end testing for user experience would be key components of this phase. Training sessions for both the administrators and end-users would also be planned and conducted during this phase.

The Unified Process model ensures adaptability and responsiveness to changes, aligning well with the dynamic needs and requirements of DriverPass. It promotes continuous communication with stakeholders, iterative development, and feedback integration, which are crucial aspects for a project of this nature.

## Object Model Application

In applying an object model to design the DriverPass scenario, an appropriate approach would be the Unified Modeling Language (UML), which is a widely-used standard for modeling object-oriented systems. To start, I would identify key objects within the system based on the requirements outlined during the client meeting. Objects such as 'User', 'DrivingLesson', 'Package', 'Reservation', 'Exam', and 'Payment' would likely be central components. These objects encapsulate the various data and functionalities needed to represent and manage the different aspects of DriverPass services.

For instance, the 'User' object would encapsulate attributes like 'first name', 'last name', 'phone number', 'address', and 'username'. Methods associated with this object could include functionalities like 'changePassword()' and 'updateContactDetails()'. Similar features like "lesson time", "lesson date", "car details", and "driver details" would be contained in the "DrivingLesson" object. 'updateLessonDetails()', 'scheduleLesson()', and 'cancelLesson()'  are possible methods for this object.

Using UML, relationships between items would also be represented, outlining how they interact and work together with one another. A 'User' object, for instance, may be linked to a 'Reservation' object, indicating that a user can book driving lessons.

With regard to inheritance, a hierarchy of packages might be depicted, with a general "Package" object being extended by "PackageOne," "PackageTwo," and "PackageThree" objects to reflect the various levels of service provided.

The usage of an object model, such as UML, provides for a visual depiction of the system's structure and behavior. It explains how various objects relate to one another, their attributes, and what operations they may execute. This makes it easier to confirm that the design satisfies the client's needs and promotes effective interaction between development teams and stakeholders. The system design is also made more scalable, maintainable, and reusable when an object model is used.

## Process and Object Model Comparison

For the DriverPass scenario, it is critical to compare the benefits and drawbacks of the object model (Unified Modeling Language - UML) with the process model (Unified Process). The Unified Process (UP), starting with the process model, has a number of benefits. First of all, UP is progressive and iterative, allowing for adaptability and flexibility to shifting requirements. Given that DriverPass' project comprises a variety of stakeholders with changing requirements and preferences, this is very advantageous for them. The iterative process guarantees ongoing system improvement and validation, which is in line with the project's dynamic character.

In addition, UP places a strong emphasis on risk management and mitigation, which is essential for a project like DriverPass where uncertainties are widespread. UP assists in reducing the impact of possible problems and enabling smoother development by identifying and managing risks early in the project. The different UP stages (inception, elaboration, building, and transition) offer discrete benchmarks and objectives, facilitating efficient project management and monitoring development.

UP, however, is not without flaws. It may require a lot of resources, a committed staff, and significant documentation. This might pose challenges for smaller projects or those with limited resources. Furthermore, the iterative process occasionally results in scope creep if it is not well controlled, which might affect schedules and costs.

On the other hand, the object model, particularly UML, provides a standardized and well-recognized method of visualizing system components and interactions. UML provides a standard language for developers and stakeholders to communicate and understand the structure and behavior of the system. This transparency is essential for a project like DriverPass where stakeholder participation is essential. The drawback of UML is its potential for complexity, particularly in bigger projects. It can take a lot of effort to create and maintain thorough UML diagrams, which could be overwhelming for team members who aren't UML experts.