|  |  |
| --- | --- |
| **Academic Year: 2020-21** | |
| **SUBJECT: Software Engineering CLASS:** SE (I)  **SEMESTER:** IV | |
| **ERP ROLL NO**: 34  **Teams Sr. No:** 20 | **NAME:** Kaustubh Shrikant Kabra |
| **ASSIGNMENT NO**.: 1 | **DATE OF SUBMISSION**:28/03/2021 |

**Title of assignment:** Select appropriate SDLC for your mini project and justify it.

**Theory:**

Drowsiness is a serious concern when driving and can cause accidents because it impairs the elements of human performance that are critical to safe driving: slower reaction time, reduced vigilance, deficits in information processing.

Existing drowsiness detection methods include:

* Carnegie-Mellon Research Institute:  
  PERCLOS Systems\_PERCLOS (percentage closure) is defined as the measurement of the percentage of time the pupils of the eyes are 80% or more occluded over a specified time interval. It has been found that PERCLOS is a reliable measure in detecting drowsiness.
* Head position metrics:  
  Systems have been devised such that the head position of the driver is detected and when the head leaves the headrest past a certain threshold percentage, the system alerts the driver.

The purpose of the drowsiness detection system is to aid in the prevention of accidents passenger and commercial vehicles. The system will detect the early symptoms of drowsiness before the driver has fully lost all attentiveness and warn the driver that they are no longer capable of operating the vehicle safely. This device will not, however, guarantee that the driver will be fully awakened and that an accident will be avoided. It is simply a tool for improving driver safety; focusing primarily on long-haul truck drivers, nighttime drivers, people driving long distances alone or people suffering from sleep deprivation.

The Systems Development Life Cycle (SDLC) gives structure to the challenges of transitioning from the beginning to the end of your project without forgetting a step.A number of different SDLC methodologies are used today to guide professionals through their project-based work. Some Models To look out….

1. Waterfall Model

[Waterfall](http://seowebsitedesign.com/wp-content/uploads/2012/12/waterfall-model-software.png) is the oldest and most straightforward of the structured SDLC methodologies — finish one phase, then move on to the next. No going back. Each stage relies on information from the previous stage and has its own project plan. Waterfall is easy to understand and simple to manage.

But early delays can throw off the entire project timeline. And since there is little room for revisions once a stage is completed, problems can’t be fixed until you get to the maintenance stage. This model doesn’t work well if flexibility is needed or if the project is long term and ongoing.

2. V-Shaped Model

Also known as the Verification and Validation model, the V-shaped model grew out of Waterfall and is characterized by a corresponding testing phase for each development stage. Like Waterfall, each stage begins only after the previous one has ended.

This model is useful when there are no unknown requirements, as it’s still difficult to go back and make changes.

3. Iterative Model

The Iterative model is repetition incarnate. Instead of starting with fully known requirements, you implement a set of software requirements, then test, evaluate and pinpoint further requirements. A new version of the software is produced with each phase, or iteration. Rinse and repeat until the complete system is ready.

One advantage over other SDLC methodologies: This model gives you a working version early in the process and makes it less expensive to implement changes. One disadvantage: Resources can quickly be eaten up by repeating the process again and again.

4. Spiral Model

One of the most flexible SDLC methodologies, the [Spiral model](http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/) takes a cue from the Iterative model and its repetition; the project passes through four phases over and over in a “spiral” until completed, allowing for multiple rounds of refinement.

This model allows for the building of a highly customized product, and user feedback can be incorporated from early on in the project. But the risk you run is creating a never-ending spiral for a project that goes on and on.

5. Big Bang Model

A bit of an anomaly among SDLC methodologies, the Big Bang model follows no specific process, and very little time is spent on planning. The majority of resources are thrown toward development, and even the client may not have a solid grasp of the requirements. This is one of the SDLC methodologies typically used for small projects with only one or two [software engineers](https://www.roberthalf.com.au/blog/software-engineer-vs-systems-engineer-whats-difference).

Big Bang is not recommended for large or complex projects, as it’s a high-risk model; if the requirements are misunderstood in the beginning, you could get to the end and realize the project may have to be started all over again.

6. Agile Model

By breaking the product into cycles, the [Agile model](http://istqbexamcertification.com/wp-content/uploads/2012/01/Agile-model.jpg) quickly delivers a working product and is considered a very realistic development approach. The model produces ongoing releases, each with small, incremental changes from the previous release. At each iteration, the product is tested.

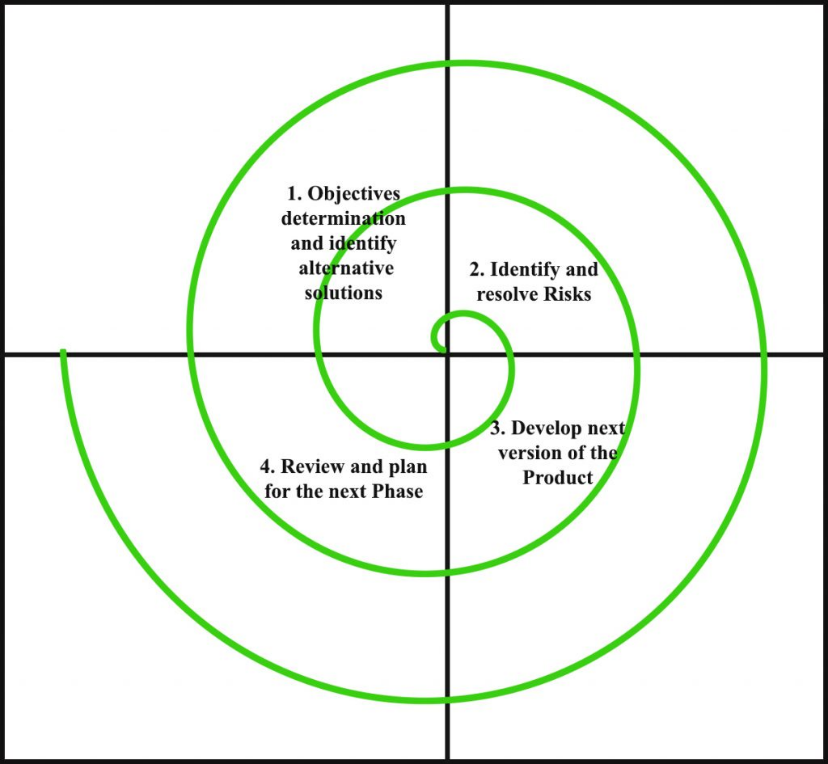
This model emphasizes interaction, as the customers, developers and testers work together throughout the project. But since this model depends heavily on customer interaction, the project can head the wrong way if the customer is not clear on the direction he or she wants to go.

**Software Development Model:**

**Spiral Model**

One of the most flexible SDLC methodologies, the spiral model takes a cue from the Iterative model and its repetition; the project passes through four phases over and over in a “spiral” until completed, allowing for multiple rounds of refinement.

This model allows for the building of a highly customized product, and user feedback can be incorporated from early on in the project. But the risk you run is creating a never-ending spiral for a project that goes on and on.



**Requirements**

* Reliability: The solution should reliably detect drowsiness so that it can serve its purpose as a system for promoting driver safety.
* Real-time response: The operation of a vehicle can involve relatively high speeds, a system that cannot detect drowsiness and warn that driver promptly can lead to serious consequences.
* Unobtrusive: It is very important that the solution is as transparent to the driver as possible.
* Economical: Existing solutions to this problem are available today but the effective ones are usually too expensive for widespread implementation.
* Flexible: To be effective, the solution should be designed so as to accommodate for all types of users, in terms of physical attributes.

**Constraints**

* Space: The solution needs to be implemented in a space-efficient manner. It must not interfere with the existing controls of the car.
* Power: There will be a limited power source so the solution needs to designed so that it can operate properly on limited power requirements.

**Proposed design**

* Eye detection algorithm:  
  The physiological properties and appearances of the eyes will be investigated and the method of capturing these properties of the eyes using infrared lighting will be explored. Kalman trackers will be used to determine eyes and head dynamics between successive images and a probabilistic model will be used to calculate the driver’s vigilance.
* Lane tracking detection algorithm:  
  Using image processing techniques measures the behavior of a driven vehicle with respect to the vehicle’s position to the surroundings.

**Conclusion:** The purpose of the drowsiness detection system is to aid in the prevention of accidents passenger and commercial vehicles. The system will detect the early symptoms of drowsiness before the driver has fully lost all attentiveness and warn the driver. It will not only help the driver but also the passengers and pedestrians in the surrounding.