**1. Waterfall Method:**

**Process**:

Requirements analysis, software design, programming, software testing, operation and maintenance;

Priorities: We should focus more on the process of requirements analysis, software design, because it’s hard for this model to make a change in the end of the project which will cost a lot!

**Advs**:

1). It plan everything in sufficient detail so people can get it right the first time

2) This method provide a template that allows analysis, design, coding, testing, and support methods to have a common guide under the template.

**Dis-advs:**

1). It will generate lots of documents

2). This model is not adapt to changes in user needs.

3) Since the development model is linear, users can only see development results until the end of the process and this will increase the risk of development.

4). Early errors may not be discovered until the later stages of development and this will lead to serious consequences.

5). For the **people’s security** during experimentation: The task also has high risk, e.g. people may be injured or killed if the solution doesn't work properly. If we want to improve the security for the system, we must test many times and change many designs, this will not be adapt by Waterfall Method

**Size of team:**

Medium-sized projects (7-40 people)

**2. Rational Unified Process (RUP)**

**Process**:

Inception, Elaboration, Construction, Transition

Priorities: I think elaboration is the most important phase. This phase can mitigate risks by elaboration of use case model and design of software architecture.

**Advs**:

1). Based on develop software iteratively:

1>. Reduce the risk for this project

2>. Determine demand in advance

3>. Build a subset of requirements

4>. Work sequentially or in parallel with other developers

5>. Users can change their requirement

6>. Suitable for large projects that can be developed in parallel

7>. Provide customers with early functionality for review

\* Based on all of the above advantages, RUP can much more effectively than waterfall method in avoiding the risk of experimentation. People can test the hardware and cods iteratively to reduce the risk.

2). Based on component-based architecture

1>. Bottom-up design, implementation, and testing of the architecture

2>. Architecture is flexible to adapt to the changes

3). Based on visually model software: It use some structure and behavior in Unified Modeling Language (UML) to helps to visualize the system and interactions

4). Verify software quality:

1>. Make verification and validation be the part of the process and get improvements on reliability, functionality, performance and **people’s security**

2>. Test for every iteration to improve the security of experimentation

**Dis-advs:** RUP only covers the development process. It does not completely cover the software process, it loses two important phases of maintenance and technical support.

**Team size:**

Large projects (> 40 people)

**3. eXtreme Programming**

**Process**: Planning, design, coding, test

Priorities: I think design is most important in this method, because during the project, programmers will test the codes and function very frequently (every week, every day, every stage) and they try their best to get a high quality code. This will benefit for the security in the experimentation

**Advs**:

1). User can change their needs whenever they want

2). Can reduce the risk during the project efficiently

3). Programmer will test the code and hardware every time even every day

4). It has small, new and valuable releases

5). It can **successfully runs all the tests** and has no duplicated logic

6).It can always find a series of small, low-risk steps

7). Always two people work together so they can much easier think of new test

**Dis-advs:**

1). This method is only for small teams, if this project need more techniques and skills, it’s hard to finish the project

2). Two people write a code at the same time. If they have differences, this will be a big poblem.

**Team size:**

Small projects (< 7 people)

**Summary:**

For this lane change problem, waterfall method is not suitable and for RUP and XP, I think XP is more suitable. Because RUP only covers the development process, it loses two important phases of maintenance and technical support and for this project, maintenance and technical support is equal important. Using XP, we can test every day, we can test every hardware in every stage. This can guarantee the safety of the experimentation and avoid hurt or kill person more effectively.