

Lifecycle Models

• Case Study 1: Mobile Game

The best approach to develop this game would be by following the evolutionary prototyping model since they have a small timeframe to deliver the project and it is a small company as well (like many startups do nowadays). Even though the agile methodologies (like scrum) is not mentioned in the lifecycle models, the evolutionary prototyping is what gets closer to such methodologies in which they need to deliver in constant phases (or sprints) providing “micro-upgrades” to the testing team, with a product that would need to work on the latest and greatest devices. Due to the nature of the perceptual computing sensors they will make use of and the one the company is targeting, combining the previous model with a staged delivery would be more than optimal, so they can perform regression testing and rollback in case something goes wrong with new hardware that hasn't been tested out yet and is completely out of the scope of the project itself.

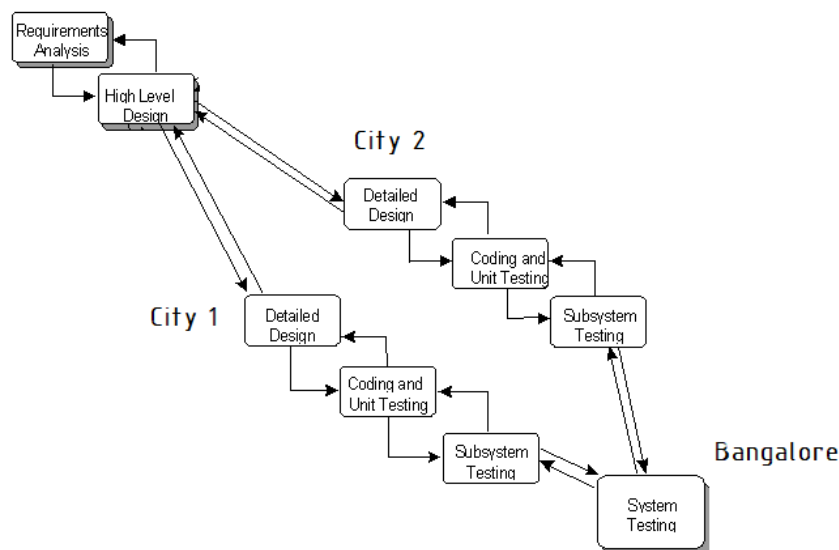
It is also worth considering that the testing phase is contemplated within the sub-stages of the staged delivery cycle and since the testing services are located in a different location from where the development team is, this doesn't represent a risk because the time zone between the two countries (Russia and Norway) is barely noticeable, just two hours between one another, so in case of any support requirements the cloud platform has, this would take place in no time and it definitely helps mitigate the risks of delaying the project even for one day. Waiting until next day to solve any sort of issues during the testing phases the team in Norway may encounter could delay the 3-month deadline.



• Case Study 2: X-Ray Diagnostic Machine

The company has identified the need to provide additional software for the product that they produce.

According to the deadline of 6 months, and that the Team is distributed in 3 cities far away from each other (2 for development and 1 for testing), the best approach to follow is based on a waterfall model with subprojects.

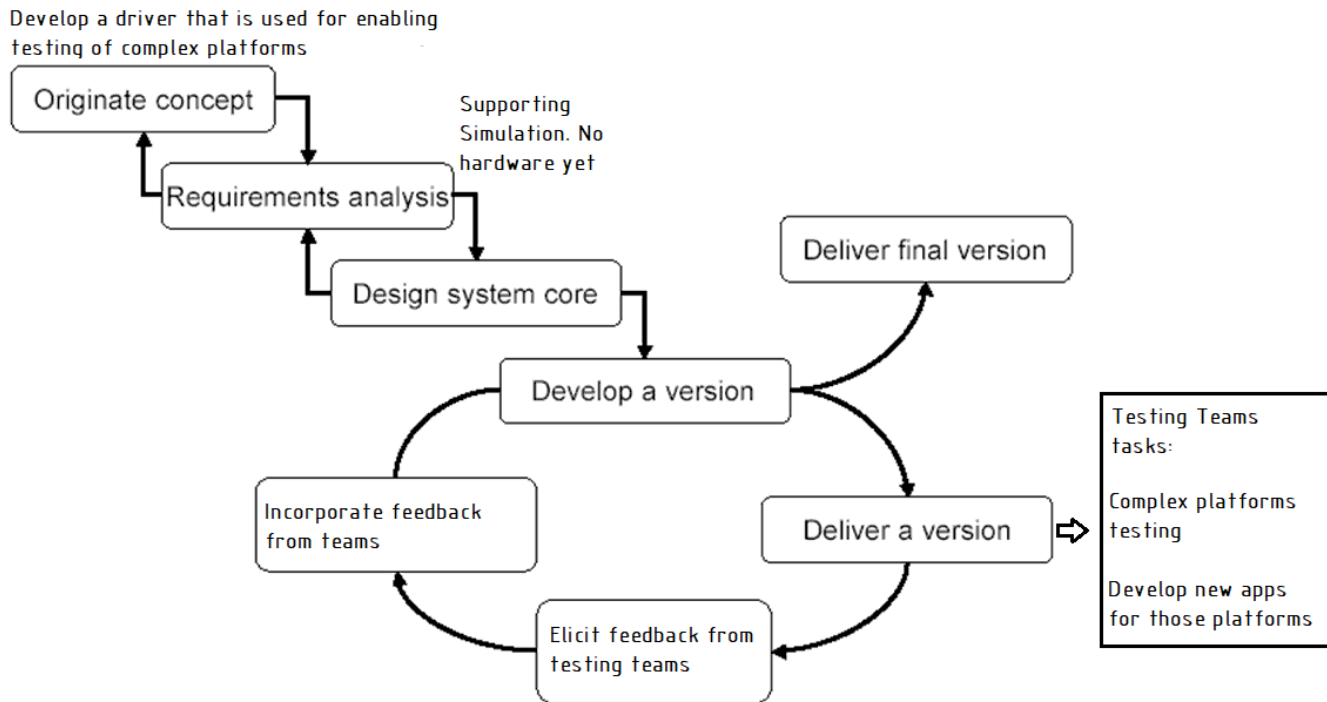


- **Case Study 3: Research Group**

[illegible]

- **Case Study 4: Graphical Driver**

Model proposed: Evolutionary Delivery



- **Case Study 5: Financial System**

Banking and financial activities always represent a big risk within the software development process because handling these kinds of sensitive data can lead to potential errors and customer complains when the software is in production. Considering this, the *Spiral model* would be the best approach to work on the development of this module. It is also worth mentioning, that, since the development of this, would be outsourced, a lot of information from the Bank itself would be in the hands of the developers/sysadmins/dbas so the small company working on it needs to take special attention to every iteration they perform following this life cycle model.

In the end the used model helps identify and reduce risks and evaluate any possible alternatives to take in case something goes wrong just as the model states, they develop the deliverables for each iteration and verify that no sensitive data has been altered or misused.