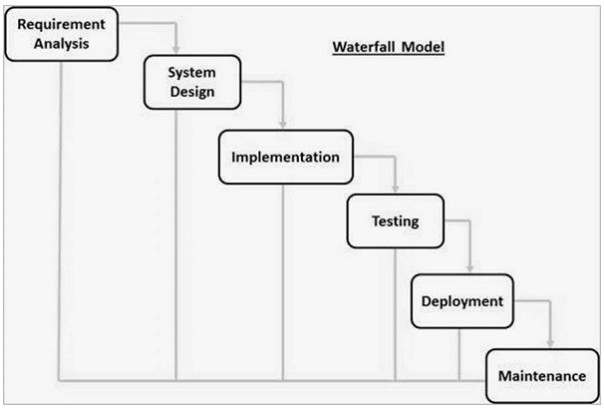
MK Coding

Risk Management in Lifecycle Models

WATERFALL MODEL



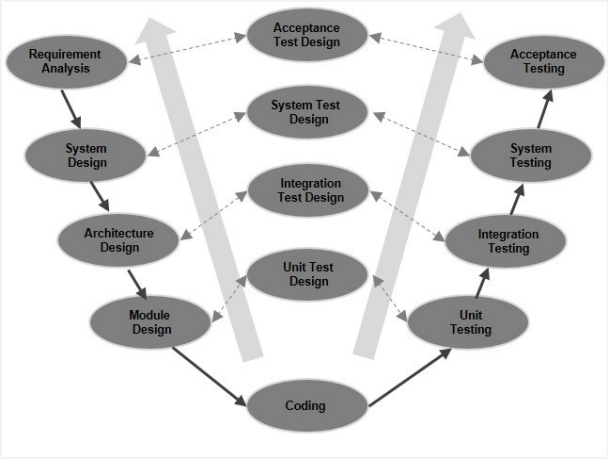
(Tutorials Point, n.d.)

Waterfall relies on having a single line from start to finish that accounts for many potential steps in the delivery of a project. This means that each step is carried out sequentially, as is the case in other linear models. The waterfall model relies on rolling back to the previous step in order to correct issues, which is the primary method of risk management in this model. This allows for issues to be handled by re-planning or re-executing the previous step with the knowledge of what an issue is and using a problem-solving method or identification system to catalogue any issues and correct for them.

As well as this, the waterfall model allows for a small level of iteration, like in the case of a requirement not being factored in at the start of the process. This model can be reused to integrate new requirements, involving a new phase of system design and implementation including the new requirements.

Some of the issues with this model are that the lack of review during design and implementation means that it can be at risk of losing sight of the original objectives. Implementing an iterative system that involves performing each step while evaluating the implementation of said step and analysing how it could be improved could easily make the waterfall model much safer in terms of risk management.

V-MODEL



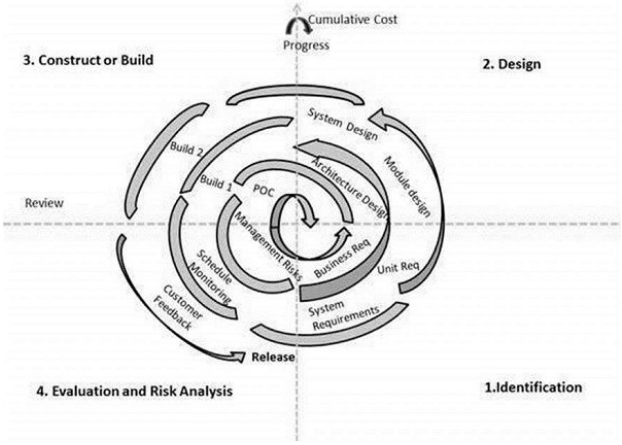
(Tutorials Point, n.d.)

The v-model is like the waterfall model, in that it's linear and relies on going back (in this case to the left rather than truly back to the last step) in order to remedy and issues. This model contains specific steps for designing how each later step will be performed and reviewed, allowing for the simplistic optimisation and problem solving that the waterfall method has minor issues with.

The v-model could also do with having an iterative section like with the waterfall model, however it's clear from the diagram that design and evaluation are options at any point in the process. Having the clear option for review and redesign affords this model a clear option for risk management and prevention.

Again, some of the issues with risk management in this model arise from the lack of a specific review phase and frequent requirement analysis. This is less of a problem than in the waterfall model due to the specific steps and paths for redesign, however there's no explicit instruction to review and analyse what's been produced. Having this extra bit of complexity allows for any issues to be identified early, preventing the project from being put back a few steps and redesigned or re-evaluated.

SPIRAL MODEL



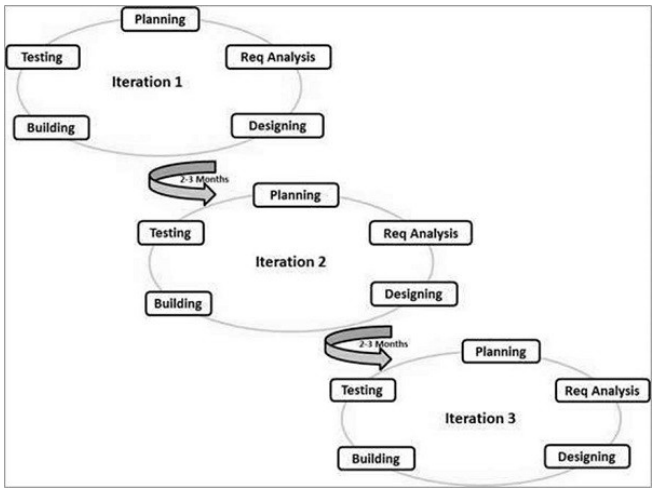
(Tutorials Point, n.d.)

While linear models may be useful for their simplicity to implement, the spiral model features a simplistic iterative procedure, that involves completing many steps in each iteration, with many iterations before a project is complete. This allows for easy prototyping and lots of review, with any potential issues being identified and caught by either the end of each iteration or a subsequent iteration.

These multiple phases allow for focussing resources on each phase individually, converting design time into review and risk analysis time. This allows for a smaller team to work multiple different roles, with intimate knowledge of the design, requirements, and construction and build methods. This prevents miscommunication and misidentification of intent, reducing the risk of requirement gathering, building, and evaluation.

As well as this, having the specified step for evaluation and risk analysis on each iteration allows for frequent review of what's been implemented, resulting in lots of opportunity for any issues to be caught and remedied in the next iteration's identification step. As well as this, the risk analysis included in this step allows for identification of goals and feasibility identification, which prevents overreach and lowers the risk of any given project.

AGILE MODEL



(Tutorials Point, n.d.)

Agile links back with the previous model, spiral, as it features multiple iterations that have specific steps for planning, testing and requirement analysis. This model involves having iterations with an implicit number of steps for completion of each iteration, which means that it's more freeform and less restrictive than the spiral model, which is rigid in how each step is ordered and applied. As well as this, the agile model can be modified to include extra steps very easily.

The image source from Tutorials Point above outlines five steps in each iteration, namely planning, testing, requirement analysis, designing and building. These five steps cover the requirements and planning stage, as well as the testing and review. These two stages are critical to minimise risk as they allow for re-evaluation and identification of issues very easily, with re-planning allowing for a solution to these problems to be forged very quickly.

An issue with the agile model is that the steps in each iteration are of implicit length, meaning that a step could be skipped over or shortened unnecessarily, resulting in improper application of the step itself. For example, having a short testing phase could result in some issues being unidentified, which then go on unresolved and potentially ending up in the final production.

Bibliography

Tutorials Point, n.d.. *SDLC - Agile Model*. [Online]
Available at: https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm
[Accessed 18th November 2019].

Tutorials Point, n.d.. *SDLC - Agile Model*. [Online]
Available at: https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm
[Accessed 18th November 2019].

Tutorials Point, n.d.. *SDLC - V-Model*. [Online]
Available at: https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm
[Accessed 18th November 2019].

Tutorials Point, n.d.. *SDLC - Waterfall Model*. [Online]
Available at: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm
[Accessed 18th November 2019].