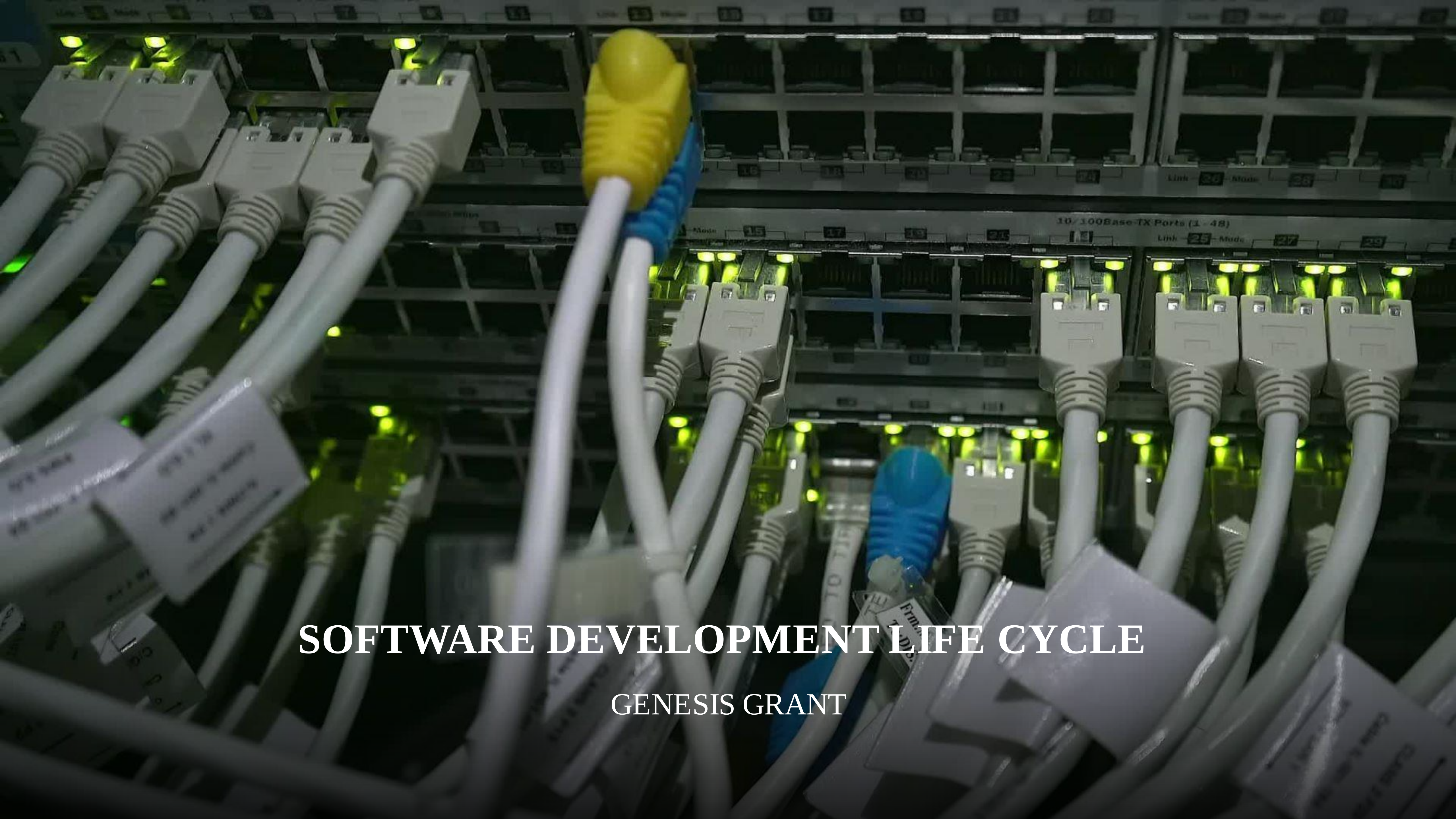


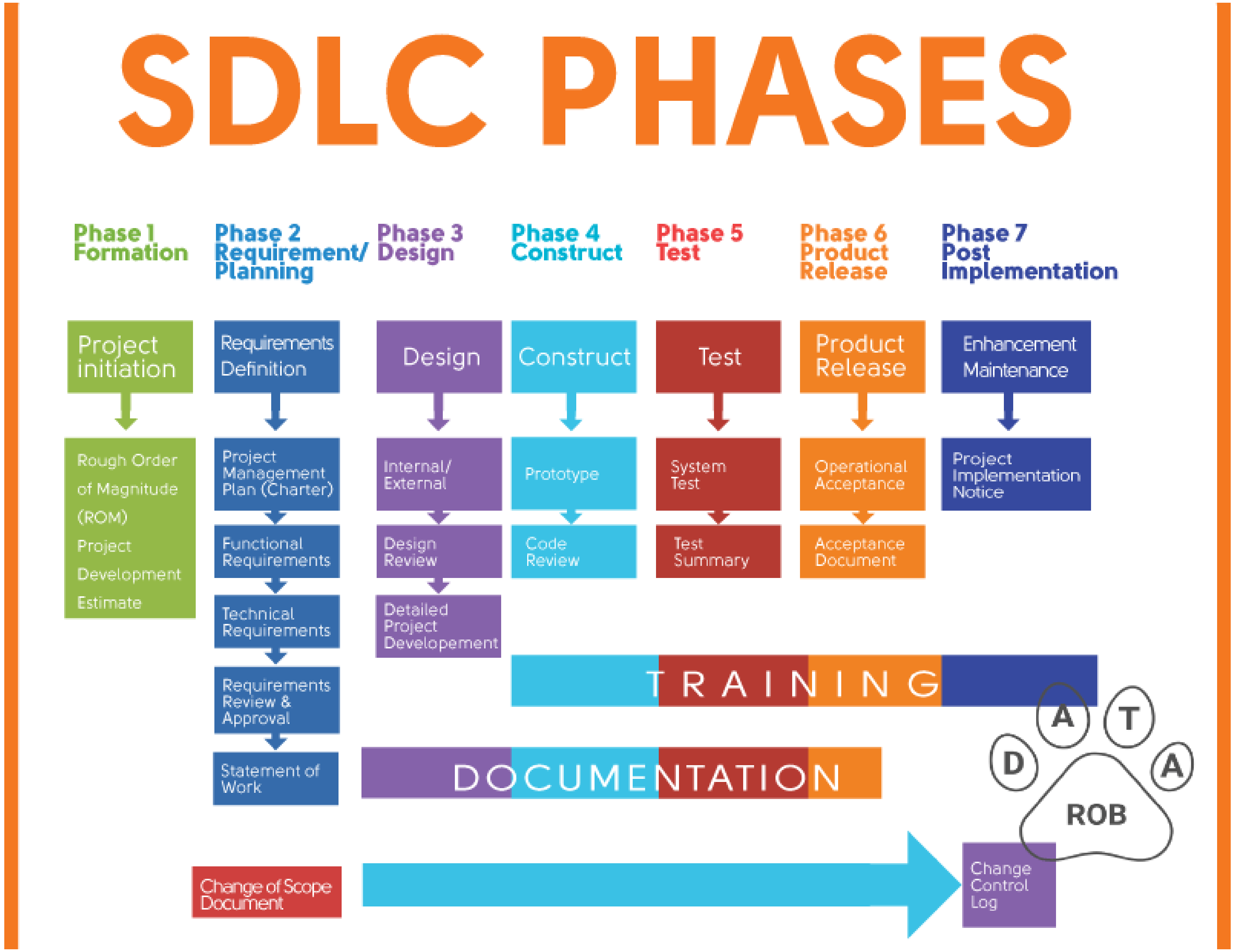
SOFTWARE DEVELOPMENT LIFE CYCLE

GENESIS GRANT

WHAT IS THE SOFTWARE DEVELOPMENT LIFE CYCLE?

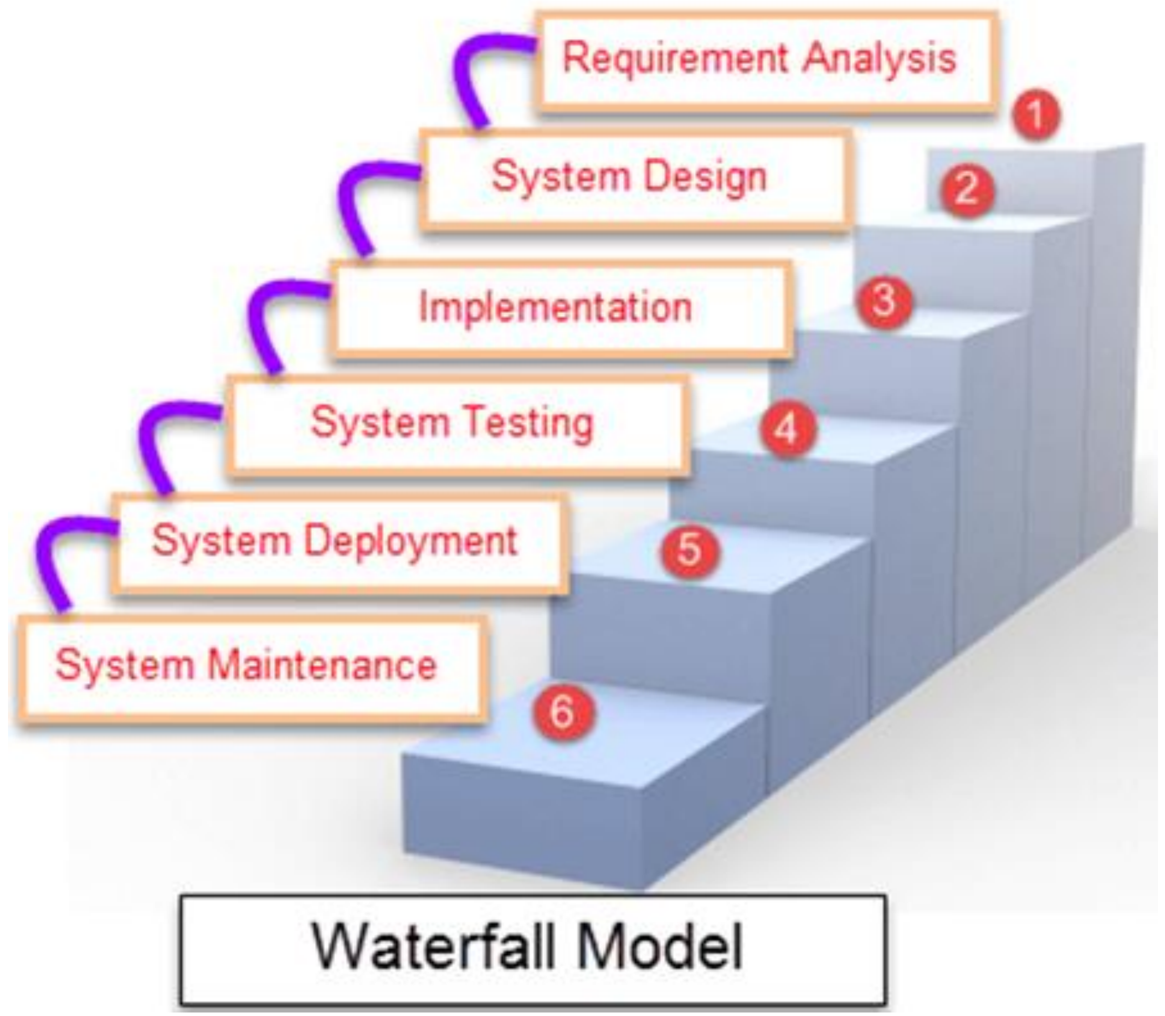
- The Software Development Life Cycle is the rubric or flow that software engineers and programmers use to draft, develop and deploy potentially high quality programs that will be used on a large population. Because it will be widely used, there cannot be many mistakes or loopholes as this can cause financial, publicity or security concerns.

- Each phase is equally vital to the development of a software.
- It takes a team to go through each process and intricately analyze a software for mistakes, errors, potential problems, etc.



WHAT IS THE WATERFALL MODEL?

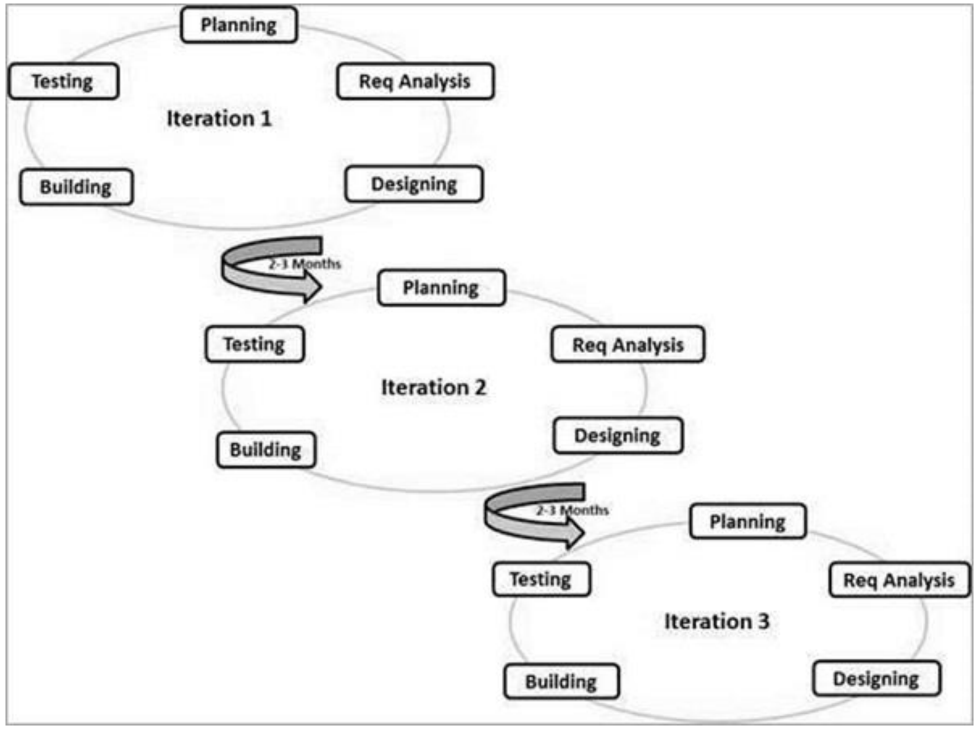
- The Waterfall Model is the SDLC but very simplified, idealistic and step by step. This model works kind of like an instruction sheet. You will follow one step then once you have the step finished and the output, your output will be the input for the next step.
- Because it is so simple, it shouldn't be used in real projects, (maybe small ones or practice models) but other SDLC models are structured after the Waterfall Model.

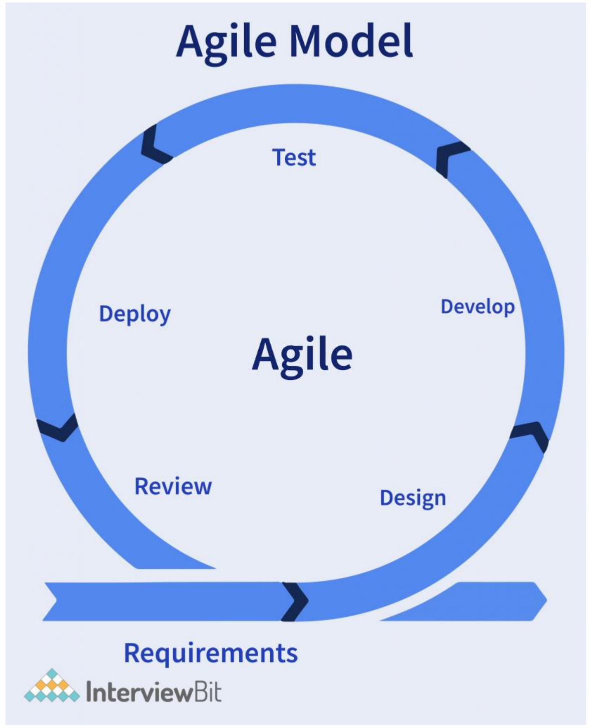


- The Waterfall model is a sequential model meaning that there is no options for changes or overlapping other phases. This can prove troublesome if there was a mistake or users would like to change a piece of the program.
- Thus again why it is useful as a rubric for other models. It is a good foundational model to help programmers and software engineers begin their development.

WHAT IS THE AGILE MODEL?

- The Agile Model, in contrast to the Waterfall Model, is based on iteration or breaking down a problem into smaller tasks with different tasks that can be assigned to different groups or individuals.





- Though the Agile Model is more prone to minimize overlooked risks and help lower project time development it can lead to other issues.
- The Agile Model can sometimes focus on individual tasks rather than the project as a whole and can lead to errors when putting it together.
- It is best used when you expect to make a lot of changes and in small projects.

WHAT IS THE DIFFERENCE BETWEEN THE AGILE MODEL AND WATERFALL MODEL?

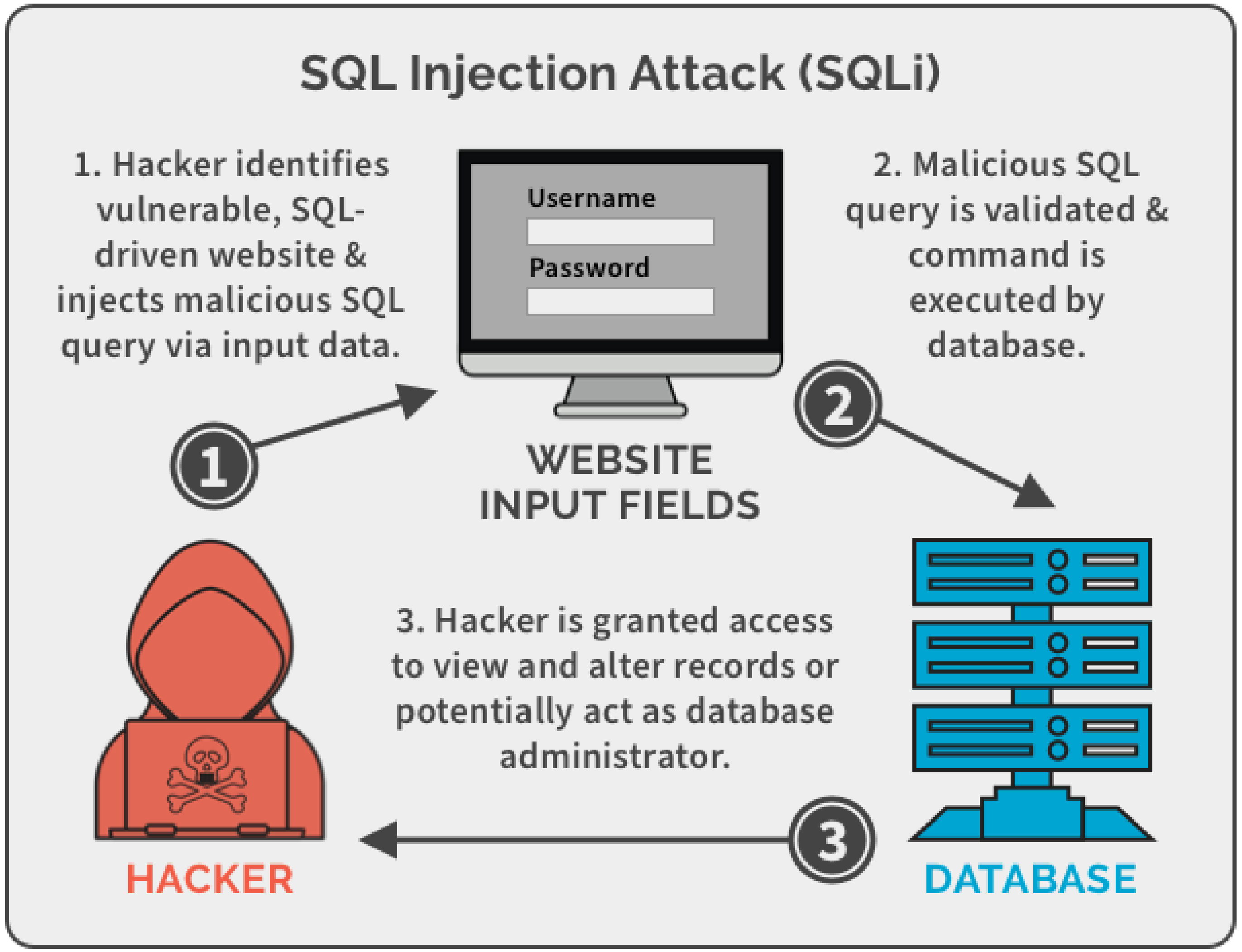
- The Waterfall Model and Agile Model are both significant rubrics to the Software Development Life Cycle.
- The Waterfall Model and Agile Model differ in their processes. The Waterfall model is a more sequential instructional process while the Agile Model is a more flexible process.
- With the Agile Model, each phase (development, review, feedback, etc.) can be answered by a simple yes or no. If no, then as developers we are able to go back and revise that step. In contrast with the waterfall model, maintenance of the project would have to be at the end.
- Neither model is better than the other; models are used based on the need of the project and preference of the developer.

WHAT ARE THE SECURITY INJECTIONS?

- Security injections are where attackers may take advantage of applications and software that are not fully secured. This can include injecting malware into the operation and forcing it to do certain commands in order to release or gain sensitive information.
- These attacks vary from SQL attacks, Command injections, code injections, etc. The common factor they have is that there were vulnerabilities the attacker took advantage of.

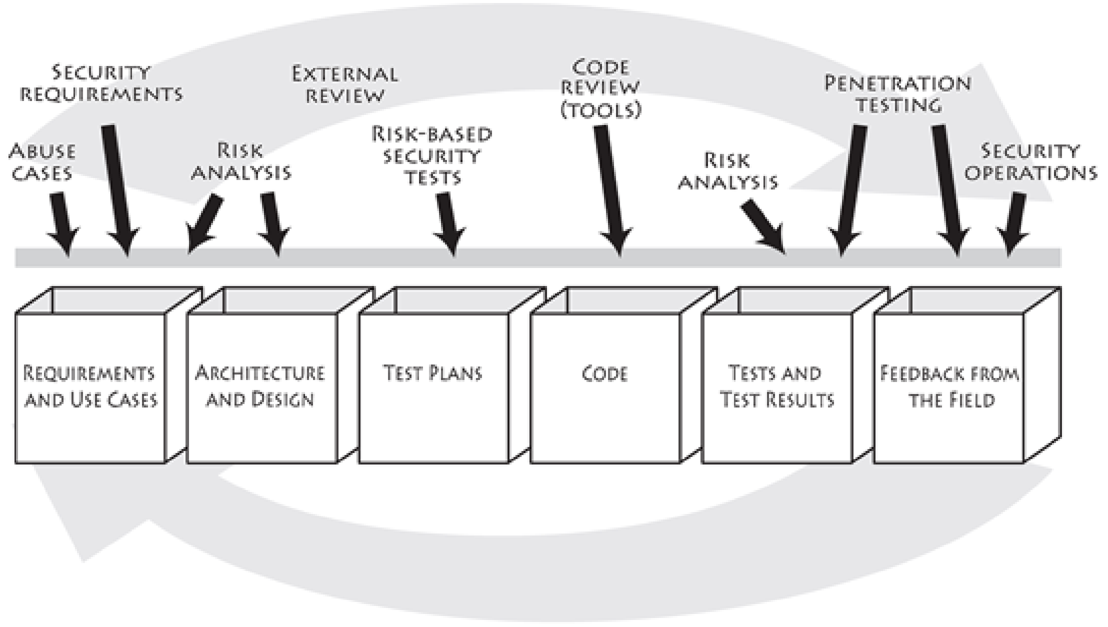
SQL ATTACKS

- This type of security injection, allows the developer to use the SQL query gain access into the site.
- Because the website was vulnerable and not entirely secured the hacker was able to get the information and do whatever they please.


SQL Injection Attack (SQLi)
1. Hacker identifies vulnerable, SQL-driven website & injects malicious SQL query via input data.
Username
Password
2. Malicious SQL query is validated & command is executed by database.
1
2
WEBSITE INPUT FIELDS
3. Hacker is granted access to view and alter records or potentially act as database administrator.
3
HACKER
DATABASE

WHAT DOES THE SOFTWARE LIFECYCLE WITH SECURITY INJECTIONS LOOK LIKE?

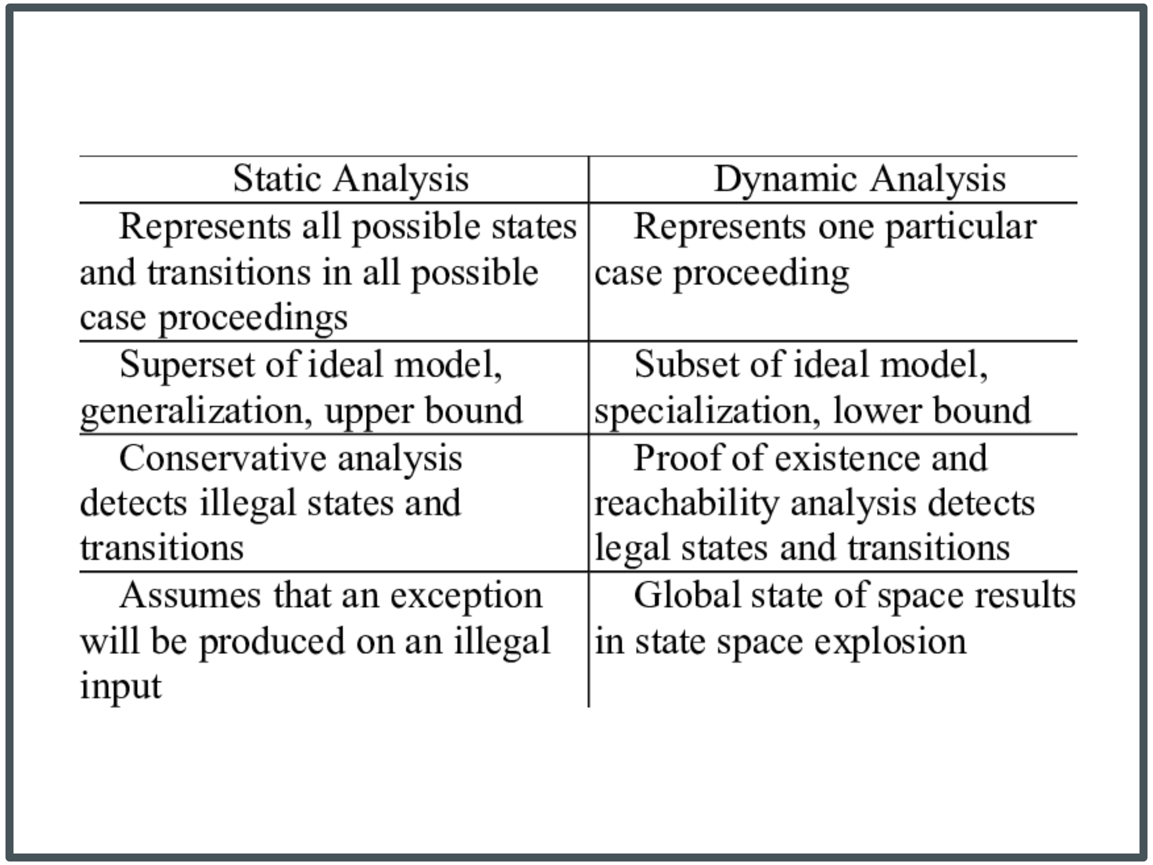
- The Software Life Cycle with security injections can look dysfunctional. Security injections and its conjoining malware, can sometimes be detrimental to programs if they are not protected. To try to find injections, you would have to continue to go back through the processes until you are able to troubleshoot the issue. To prevent security injections, developers should always keep in mind the potential security concerns throughout the cycles and plan accordingly.



- This graphic explains how security concerns should be taken into account in every step of the cycles. Although the Waterfall cycle is sequential, when you continue to go through the process, you can begin to uncover other issues that may occur further strengthening your program. The Agile cycle is iterative and through this process we are able to go back steps and search for bugs.

WHAT IS THE DIFFERENCE BETWEEN THE STATIC AND DYNAMIC SOFTWARE ANALYSIS?

- Static software analysis is a more check list like approach. Instead of running the program, developers will instead proofread code and make sure the foundation does not have any issues.
- Dynamic software analysis is a more test trial approach. The developers will run the program and examine any errors that may have been overlooked. The primary role is to find any bugs and secure the program furthermore.



Static Analysis	Dynamic Analysis
Represents all possible states and transitions in all possible case proceedings	Represents one particular case proceeding
Superset of ideal model, generalization, upper bound	Subset of ideal model, specialization, lower bound
Conservative analysis detects illegal states and transitions	Proof of existence and reachability analysis detects legal states and transitions
Assumes that an exception will be produced on an illegal input	Global state of space results in state space explosion

- Static Analysis and Dynamic Analysis, allows for parallel operations to be checked, but a great way to entirely troubleshoot a program/software, is to continuously repeat the process with differing test cases.