

# Software Development Life Cycle

ITDEV117 Team 2

# Phases of a Typical Software Development Life Cycle

# Requirements and Analysis

This is the planning phase. During this time, the clients and companies determine what they need from their developed software. They they determine what would be cost prohibitive and the desired time frame. Initial relationships are built between clients, executives and developers.

# Design

During this phase the design of the software solution is determined. This is where developers sit down with clients and hash out what the clients need and establish how these needs can be met within their previously established limits.

# Implementation and Coding

The development team gathers to put forth a plan for execution. A time frame is established with various goals along milestones to the project's completion. Team members are given tasks usually by a Development Lead and regular updates are given to ensure the project is being developed within the established time frame.

# Testing

Once the initial coding is finished quality assurance teams test the 'qual' product thoroughly for any code defects, bugs or negative user experiences. If further development is needed, the code is fixed or rewritten and goes back to quality assurance for testing.

# Deployment

Once the software is finished with quality assurance and it is deemed finished with the client's specifications in mind, it is considered ready for production. The software is usually 'rolled' into production and the client is notified that their software is now live.

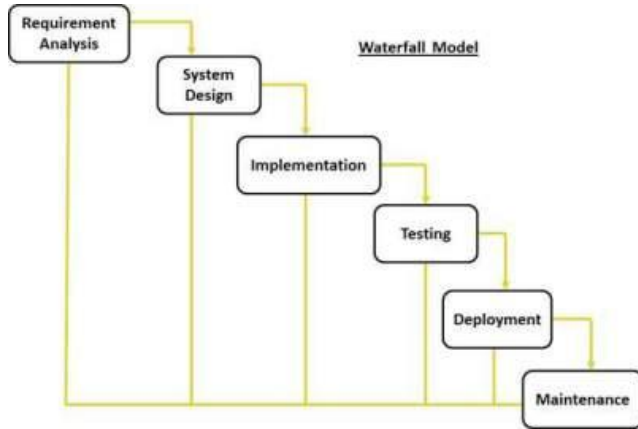
# Maintenance

Continued maintenance is required after deployment. As thorough as quality assurance may be, end users will inevitably find bugs or other issues with the software that will need to be addressed. When this occurs, the SDLC starts again with regard to the solution for a fix, culminating in a new production release.



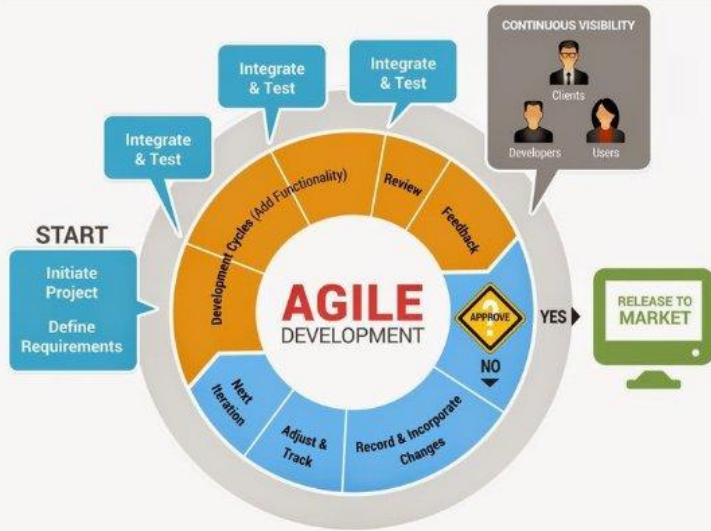
# Methodologies

# Waterfall



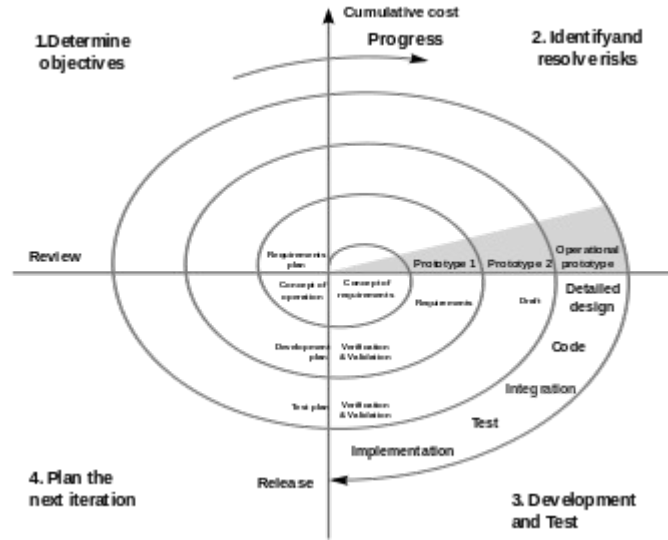
- Original Methodology
- Sequential order
- Clients only consulted once
- Potential for running over budget
- Very little flexibility for change during development

# Agile



- Most popular methodology today
- Cyclical order
- Fast-paced development cycles
- Very flexible
- Allows for quick release with less polish

# Spiral



- Preferred methodology for very complex development with a large organization
- Allows for flexibility and polish
- Most risk averse methodology
- Very slow compared to other methodologies to release

# Conclusion

There are many more methodologies than the three listed here, but these are the most common. Ultimately, every software development takes its own approach, often resulting in a unique hybrid that tailors itself to the needs of all the individuals involved. The most important reason for having a SDLC methodology is establishing a mindful approach to development to maximize efficiency and reduce present and future development costs.

# Team 2

Matthew Dal Santo

Michael Mellon

Clifton McFarlane

Felicia Jarosinski

# Cited Sources

[1]Wikipedia. Software development process, 2017,  
[https://en.wikipedia.org/wiki/Software\\_development\\_process](https://en.wikipedia.org/wiki/Software_development_process). Accessed 5 Dec. 2017.

[2]IT Knowledge portal. Software Development Methodologies, 2017,  
<http://www.itinfo.am/eng/software-development-methodologies/>. Accessed 5 Dec. 2017.

[3]TutorialsPoint. Software Development Life Cycle, 2017,  
[https://www.tutorialspoint.com/software\\_engineering/software\\_development\\_life\\_cycle.htm](https://www.tutorialspoint.com/software_engineering/software_development_life_cycle.htm). Accessed 5 Dec. 2017.

# References

BA approach to adopt Agile as a right SDLC methodology! by Ali Saghaeian  
, 2016, retrieved from <https://www.linkedin.com/pulse/ba-approach-adopt-agile-right-sdlc-methodology-ali-saghaeian/>. Copyright 2017 by LinkedIn Corporation.

TC1019 Fall 2016, What about the waterfall method? by Mike Sagnelli  
, 2016, retrieved from <https://kenscourses.com/tc1019fall2016/syndicated/what-about-the-waterfall-method/>. CC BY-SA 4.0

Spiral model - Wikipedia, 2000, retrieved from [https://en.wikipedia.org/wiki/Spiral\\_model](https://en.wikipedia.org/wiki/Spiral_model). Special Report CMU/SEI-2000-SR-008, July 2000.