

Project Management Methods

by Sophia



WHAT'S COVERED

In this lesson, you will learn about how development projects are staffed and managed through different project management methodologies. You will learn about the simple waterfall method, as well as the agile method and its derivatives, scrum and DevOps. You will also learn about the kanban methodology, which uses a different approach to managing work and workers.

Specifically, this lesson will cover the following:

1. Project Staffing and Management
 - 1a. Full-Stack Versus Specialized Development
2. Project Management Methodologies
 - 2a. Waterfall
 - 2b. Agile
 - 2c. Scrum
 - 2d. DevOps
 - 2e. Kanban

1. Project Staffing and Management

Regardless of the size of development projects, it is essential to implement an effective project management strategy to keep the project on course and aligned with its objectives. There are a few fundamental project management factors that must be taken into account at the start of a project.

1. The project's staffing requirements should be assessed to determine whether a specialized team of developers and engineers will be involved or if a single developer will suffice.
2. The project management methodology should be selected and implemented.

1a. Full-Stack Versus Specialized Development

Web development projects require skilled developers who understand their craft. The size and complexity of the project may help determine which staffing approach works best. From the project manager's perspective, the options include hiring a team of specialized developers versus hiring individual developers referred to as **full-stack developers**.

Full-stack developers refer to individual website developers with the skills and knowledge to be able to handle all aspects of a development project. Full-stack developers can handle the following:

- The front end of the project, such as designing and developing a responsive and accessible user interface
- The back-end development via server-side scripting and web server management
- Database solutions and persistent data.
- API development

Specialized developers focus on specific “specialized” skills, or expertise, of a development project. For example, a developer's skills, knowledge, and education can be focused on front-end development. That developer focuses solely on the design and build of the user interface. They are very knowledgeable about UI/UX concepts and best practices, responsive web design, as well as the compatibility concerns related to browsers and operating systems. Most projects would also require a back-end developer who would be skilled in server-side scripting and **Application Programming Interface (API)** development. There would also be a need for a data specialist who would deal with any database, persistent storage, and data warehouse and processing. To tie it all together, there would be an engineer who managed the overall architecture and design of how the website or application would be organized and how it would operate. Together, these specialized developers would focus on their areas of expertise and work closely with each other through the chosen project management methodology to develop a cohesive product.

Another effective approach, if the project resources allow for it, is to utilize **pair programming**. Pair programming refers to assigning two developers with the same skill set to a single project. The two developers work together, side by side, on a single task using a single workstation. This allows one person to write, while the other contemplates code interactions, watches for bugs, and keeps track of tasks. By altering which programmer in the pair is the lead at different intervals, both programmers are able to contribute and learn from each other. Proponents of this model highlight studies that show decreases in bugs, increased performance (it is, after all, harder to sneak in that Facebook post when coworkers are regularly using your screen), increased knowledge across staff members, and less distraction.

The success of this approach depends on the company's budget and culture. Paying two staff members to complete one task can be more expensive, possibly offsetting reduced programming time. Poor pairing decisions (e.g., two programmers with little experience) and other factors like addressing sick time and vacations that interrupt teams can also reduce the potential benefits of this approach.

TERMS TO KNOW

Full-Stack Developers

Individuals who can be involved in all aspects of a website or software development project.

Specialized Developers

An approach to software development that involves a team of individuals with specific skills, or expertise, each

focused on one aspect of a website or software development project.

Application Programming Interface (API)

A set of functions and procedures that allow programmers to create applications that interact with data and other features of a device or service, such as a website or database.

Pair Programming

A staffing approach wherein two developers are assigned to the same task and work side by side on a single workstation.

2. Project Management Methodologies

The second major consideration of any development project is the **project management methodology**. Project management is a critical component of any complex project, such as computer software and web development. Web development alone incorporates a wide range of different subject areas, technologies, and coding and programming languages. In addition to the standard elements of project management, such as managing skilled human resources, tasks, objectives, timeline, and budget, all of these factors need to be thoroughly planned, executed, monitored, and controlled throughout the project's duration. A management approach will be needed to determine the pace, goals, and deadlines and to maintain order and understanding of the project. There are a great deal of approaches to this problem, and we will take a quick look at some popular solutions.



TERM TO KNOW

Project Management Methodology

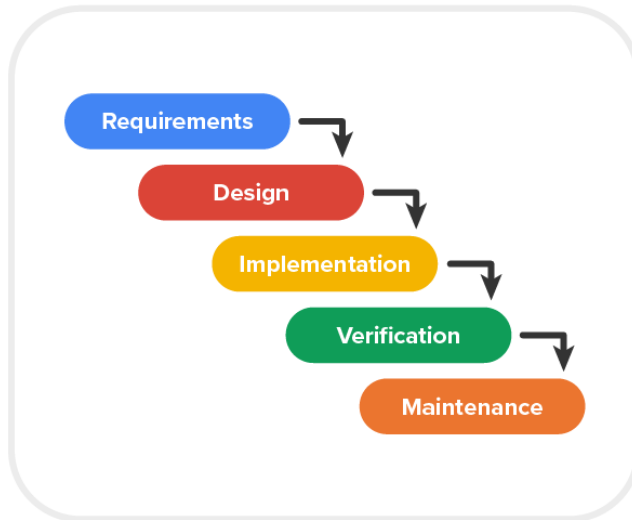
A set of guiding principles and processes used in planning, managing, and executing a project.

2a. Waterfall

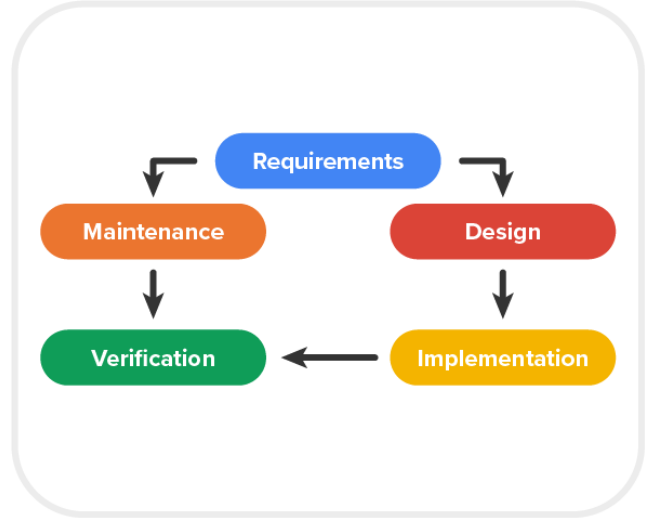
The simplest management approach is the **waterfall methodology**. Waterfall is a relatively static and linear approach to project management. The project is divided into phases that are completed in a sequential order. Waterfall is also cyclical in nature and is often presented in a circular format like the **software development life cycle (SDLC)**. Regardless of how the phases are organized, the goal is to provide a structure for continuous development and improvement of the product through a five-step process:

1. Requirements: Define the problem and set goals.
2. Design: Design how the solution will be developed and structured and what resources will be utilized.
3. Implementation: Write code and bring technology together to create the solution.
4. Verification: Once the system is working, conduct tests to ensure the solution meets the needs and is working properly.
5. Maintenance: Feedback, logs, bug reports, and feature suggestions trigger the process to begin again to resolve performance and operational issues, build new features, and resolve bugs and vulnerabilities.

Waterfall Method



Software Development Life Cycle (SDLC)



While the waterfall method is a project management methodology, it can also serve as a very high-level process that acts as a guide for some of the other methodologies that are in use today. For example, even though we may implement the next methodology called agile, the agile method is still based on the cyclical process of SDLC.



TERMS TO KNOW

Waterfall Methodology

A sequential development process that flows from one phase of the project to the next, like a waterfall, each phase being completed before the next begins.

Software Development Life Cycle (SDLC)

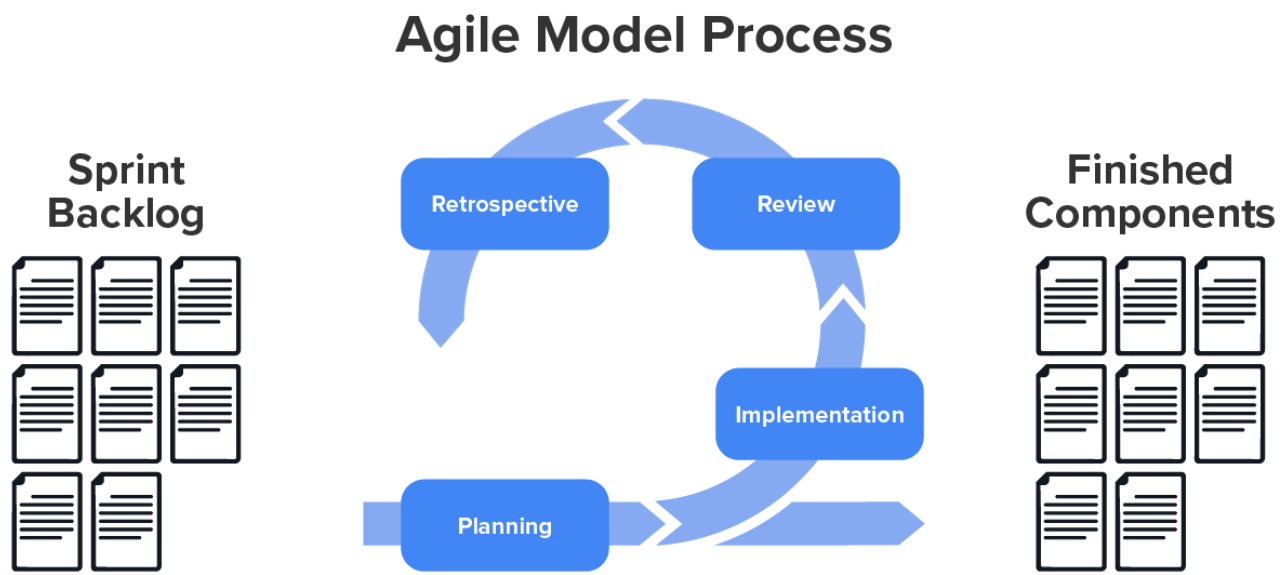
A cyclical process containing the five to seven phases of software development that is carried out repeatedly until the end of the life of the software product.

2b. Agile

The **agile methodology** of project management is one that focuses on a less rigid and less monolithic approach to the SDLC. Agile instead focuses on cross-functional collaboration that breaks larger projects down into smaller cycles called “sprints.” Sprints run concurrently, unlike waterfall, wherein each phase only takes place after the previous phase is complete, and decision-making is accomplished collaboratively, throughout the project, and across all teams. The goal of agile models is to encourage collaboration and to focus on smaller, more manageable chunks of a larger project.

The sprints are short-term goals, wherein the team collaborates on decisions and approaches to the problem. Communication is frequent among team members as well as between teams in order to keep the entire workforce up to date on the progress and changes made to the original plan. This helps ensure that teams adapt to changes and ensure that changing business needs are taken into account and incorporated into the

project. Proponents of agile argue that this methodology has the ability to produce higher-quality software and information systems.



Sprints are usually a fixed length of time, such as 1 week to 4 weeks in length, and incorporate four phases:

- Planning: A plan for the sprint goal is decided.
- Implementation: The team works toward the sprint goal.
- Review: The sprint team meets daily to review progress and communicate challenges.
- Retrospective: At the conclusion of the sprint, the team meets to discuss the successes and challenges of the sprint and the sprint team. Additionally, self-reflection takes place to identify what went well and what did not regarding the sprint team and the sprint that just concluded.

The idea of sprints is to help break a large project down into smaller sections and encourage thorough communication to ensure everyone is on the same page and that issues are worked out together as a team. The Parking Lot is another helpful concept in agile methodologies that aims to keep the team on the target and boost efficiency. With stand-up meetings being only 10–20 minutes, it is important for each member to get to the point quickly and not let the meeting, or the project, get bogged down with scope creep and nonessential topics or discussions. Instead, when a nonessential topic is introduced, one of the members of the discussion should recognize this and act quickly to “park” the idea before it eats up more time. The concept or idea gets written down and then placed in a “parking lot for ideas,” and the meeting then moves on to more important or critical topics. If there is time after the critical topics have been addressed, then the team could return to the parking lot. Otherwise, the parking lot topics are “parked” until another meeting.



To learn more about the agile Parking Lot concept, check out this brief article: [“What Is Parking Lot in Agile?”](#)

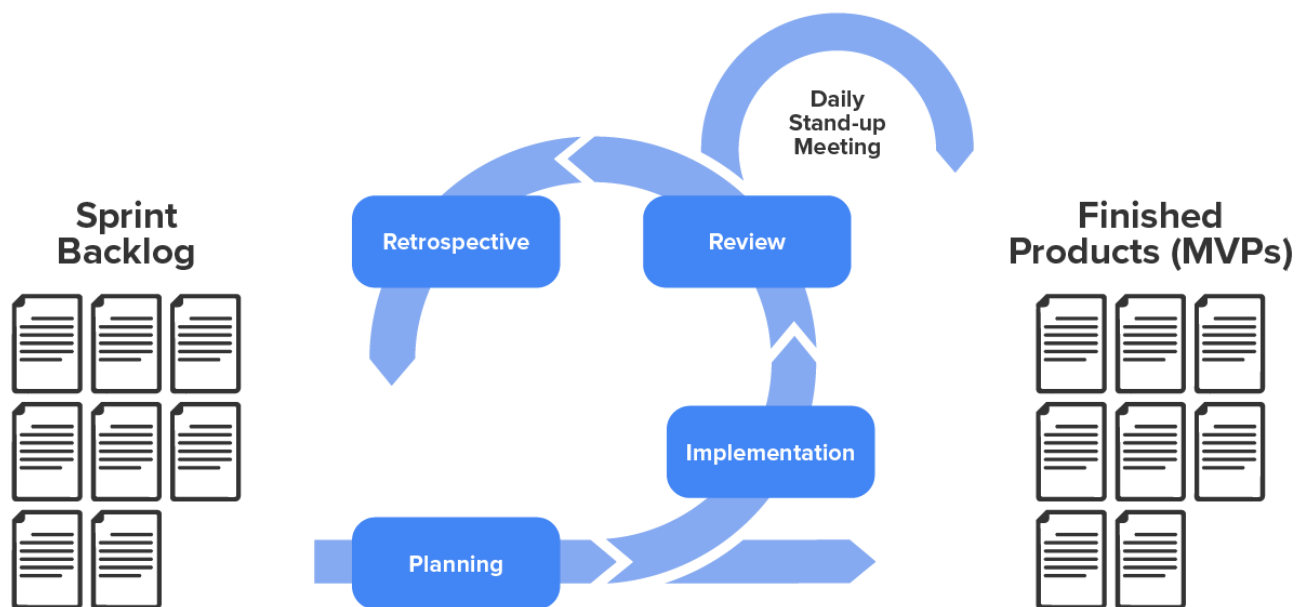


Agile Methodology

A software development process that includes requirement and solution discovery throughout the project and organizes tasks into sprints.

2c. Scrum

An extension of agile, **scrum** improves on the agile model by adding emphasis on the daily review meetings during each sprint. During the daily meetings, the sprint team conducts a brief daily review meeting, usually around 30 minutes or less, to give the sprint team a chance to communicate any issues, changes, or opportunities.



Another difference of the scrum model is that instead of just focusing on smaller goals that move closer to a final product, scrum focuses on getting customers and clients to use the products sooner rather than waiting for the finished product. This helps with customer satisfaction as they can begin taking advantage of the product. Additionally, this helps with the development process, as the sooner users can start using the product, the sooner the development team can start receiving feedback.

A small product that can be released relatively quickly is called a **minimum viable product (MVP)**. An MVP is a small working product that meets the bare minimum requirements of the product's core functionality. An MVP focuses on providing users with the core function, while less important functions and details such as UI themes, user customization, and sharing are added later.



TERMS TO KNOW

Scrum

An agile software development method that includes continuous requirement and solution discovery; however, the sprint teams are encouraged to self-manage and reflect using daily scrum meetings.

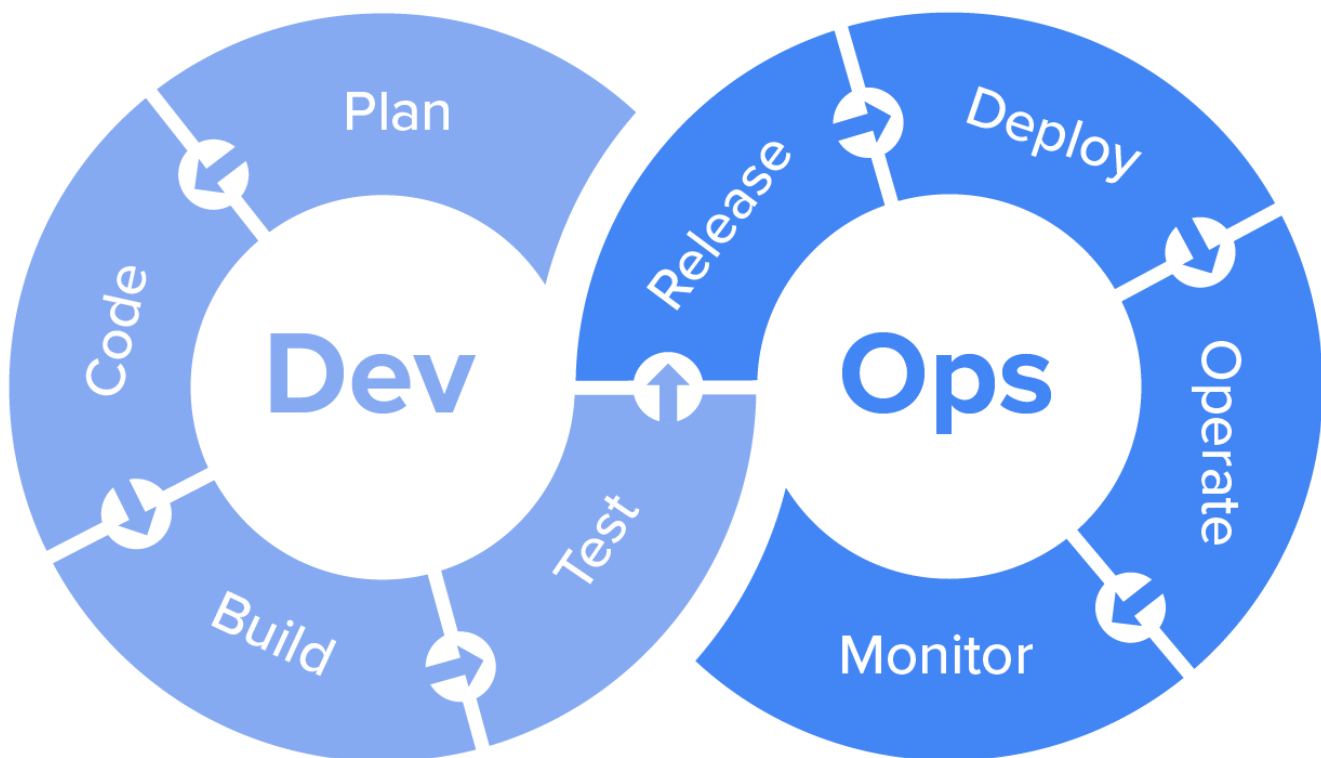
Minimum Viable Product (MVP)

A simple early version of a product that meets only the basic requirements for use and can be adapted and improved after user feedback is collected.

2d. DevOps

DevOps is another project management methodology designed for software development that focuses on communication, collaboration, and cross-functional integration of professionals with the intent of rapid production of products. The key difference is that DevOps seeks to integrate and align the goals of software development and IT operations. Furthermore, the process of feedback and workflow is automated as much as possible to ensure that feedback is gathered and quickly and directly fed into an automated process that initiates the next development task.

DevOps leverages technology and information systems to provide automation with the purpose of streamlining the flow of feedback into the development process and helping to quickly release the updated product. Automation, in the case of DevOps, aims to reduce the amount of time between receiving feedback, developing an update or product, and deploying it to the users. The integration of goals between developers and IT operations aims to increase communication and streamline the development and deployment process.



TERM TO KNOW

DevOps

A software development method that integrates the functions of IT operations with development and makes use of automation tools to reduce the time between receiving feedback and initiating a development cycle.

2e. Kanban

Kanban is a visual project management method that is derived from the Japanese word “kanban,” which means a “visual board” or a “sign.” In this method, kanban cards are created for each of the tasks that need to be accomplished and are organized on a board under different headings, such as the following:

- Backlog
- Today
- Doing
- Blocked
- Done

As items are identified, a new card is created that includes the task title, the owner, and any ongoing notes or due dates related to the task. The card is then added under the Backlog heading. Cards that are going to be tackled today are moved to the Today heading. Items being worked on at the moment are moved to the Doing heading. Any cards that cannot be completed due to a dependency on another task or some outside factor are placed under the Blocked heading with a note related to what is currently blocking the card. Lastly, completed cards are moved to the Done heading.

The benefit of this visual organization is that everyone can see what needs to be done, what is on deck to be completed today, and who is working on what as well as who has the bandwidth to work on a task.

While the act of using a kanban board is relatively simple, kanban also incorporates six key principles to guide the overall methodology:

1. Start with what you do now: While kanban is about continuous process improvement, it starts with what you currently do in order to improve existing processes.
2. Agree to pursue incremental and evolutionary change: Kanban encourages individuals and teams to make small incremental changes to improve workflow and processes.
3. Encourage acts of leadership at all levels: Everyone in the project is encouraged to take on leadership and suggest improvements based on their observations.
4. Focus on customer needs and expectations: Focus should be placed on understanding the needs and expectations of the customer in order to properly evaluate the quality of the project.
5. Manage the work, not the workers: Kanban respects the existing roles and responsibilities of the team members and empowers them to take charge of tasks, reflect on their performance, and improve upon their abilities.
6. Regularly review the network of services: Kanban encourages collaboration through the sharing of ideas, observations, and feedback regarding the various skills and services provided by the team.

Kanban also relies on six essential practices:

1. Visualizing the workflow: Creating a visual representation of the workflow helps identify bottlenecks, visualize the flow of work, and make the work more transparent.
2. Limiting work in progress: Limiting the amount of work in progress helps prevent multitasking and improve focus on completing one task at a time, thereby improving efficiency and reducing lead time.

3. Managing flow: Kanban aims to help in optimizing flow, which can be achieved by monitoring flow metrics, identifying and addressing bottlenecks, and continuously improving the workflow.
4. Making process policies explicit: Defining and communicating process policies clearly helps ensure that everyone understands how work is supposed to be done, which reduces misunderstandings and promotes consistency.
5. Implementing feedback loops: Kanban emphasizes the importance of getting feedback from customers, stakeholders, and team members as a way to identify areas for improvement.
6. Improving collaboratively: Kanban is a continuous improvement process that encourages collaboration and experimentation to identify and solve problems, improve continuously, and evolve their processes to better meet the needs of their customers.



WATCH

Check out the following video to learn more about kanban project management.



TERM TO KNOW

Kanban

Japanese for visual board or sign, and a task management methodology that uses cards to represent tasks, which are organized on a board based on their status.



SUMMARY

In this lesson, you learned about **project staffing** and methodologies for **managing** development projects. Then, you learned how an individual developer can serve as a **specialized** worker in a project or a **full-stack** developer, taking on the majority, if not all, aspects of the project. You also learned about different **project management methodologies**, including **waterfall**, **agile**, **scrum**, **DevOps**, and **kanban**.

Source: This Tutorial has been adapted from "The Missing Link: An Introduction to Web Development and Programming " by Michael Mendez. Access for free at <https://open.umn.edu/opentextbooks/textbooks/the-missing-link-an-introduction-to-web-development-and-programming>. License: **Creative Commons attribution: CC BY-NC-SA**.

REFERENCES

Kanbanize (2023). What Is Kanban? Explained for Beginners. Retrieved from <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban>



TERMS TO KNOW

Agile Methodology

A software development process that includes requirement and solution discovery throughout the project and organizes tasks into sprints.

Application Programming Interface (API)

A set of functions and procedures that allow programmers to create applications that interact with data and other features of a device or service, such as a website or database.

DevOps

A software development method that integrates the functions of IT operations with development and makes use of automation tools to reduce the time between receiving feedback and initiating a development cycle.

Full-Stack Developers

Individuals who can be involved in all aspects of a website or software development project.

Kanban

Japanese for visual board or sign, and a task management methodology that uses cards to represent tasks, which are organized on a board based on their status.

Minimum Viable Product (MVP)

A simple early version of a product that meets only the basic requirements for use and can be adapted and improved after user feedback is collected.

Pair Programming

A staffing approach wherein two developers are assigned to the same task and work side by side on a single workstation.

Project Management Methodology

A set of guiding principles and processes used in planning, managing, and executing a project.

Scrum

An agile software development method that includes continuous requirement and solution discovery; however, the sprint teams are encouraged to self-manage and reflect using daily scrum meetings.

Software Development Life Cycle (SDLC)

A cyclical process containing the five to seven phases of software development that is carried out repeatedly until the end of the life of the software product.

Specialized Developers

An approach to software development that involves a team of individuals with specific skills, or expertise, each focused on one aspect of a website or software development project.

Waterfall Methodology

A sequential development process that flows from one phase of the project to the next, like a waterfall, each phase being completed before the next begins.