

Software Development Methodologies

by Sophia



WHAT'S COVERED

As information systems have evolved, so has the process by which new software applications are developed. Recall that today's software applications are developed in a series of steps, or phases, based on particular methodologies. In this tutorial, we will discuss the various software development methodologies.

Our discussion will break down as follows:

1. Software Development Methodologies

Due to the complexity associated with the process of developing software applications, developers and engineers apply various methodologies to the development process. Depending on factors such as project scope, complexity, time, change occurrence, and team size, development teams may decide on a particular methodology. A **software development methodology** represents an attempt to split the process of software development into separate phases. By breaking the entire process down into phases, engineers are able to better plan and manage the software development process. Recall that the software development life cycle (SDLC) provides a general list of the phases that should be contained within a software development methodology. In fact, a software development methodology is considered to be a component of the much broader software development life cycle. The table below lists some of the common methodologies.

SDLC Methodology	Best For
Cascade	--Situations in which testing is very critical and happens frequently
Rapid Application Development (RAD)	--Situations in which development resources need to be tightly controlled --Situations in which development needs to occur in a timely manner --Risks need to be controlled
Agile	--Situations in which the development team collaborates well --Situations in which the client needs to be highly involved with the project
Lean	--Situations in which time and other resources need to be managed --Development team is technically proficient
	--Situations in which feedback is needed frequently

IN CONTEXT

Bison Software is currently working on its latest software project, Game Creator. Game Creator will give any person equipped with an iOS device, such as an iPhone or iPad, the ability to create games and other forms of interactive media from their device. Any form of media created within Game Creator can then be easily uploaded to Bison Software's Game Creator community section of the company's website for testing.

Let's take a look at how the various software development methodologies can be applied to the Game Creator project.

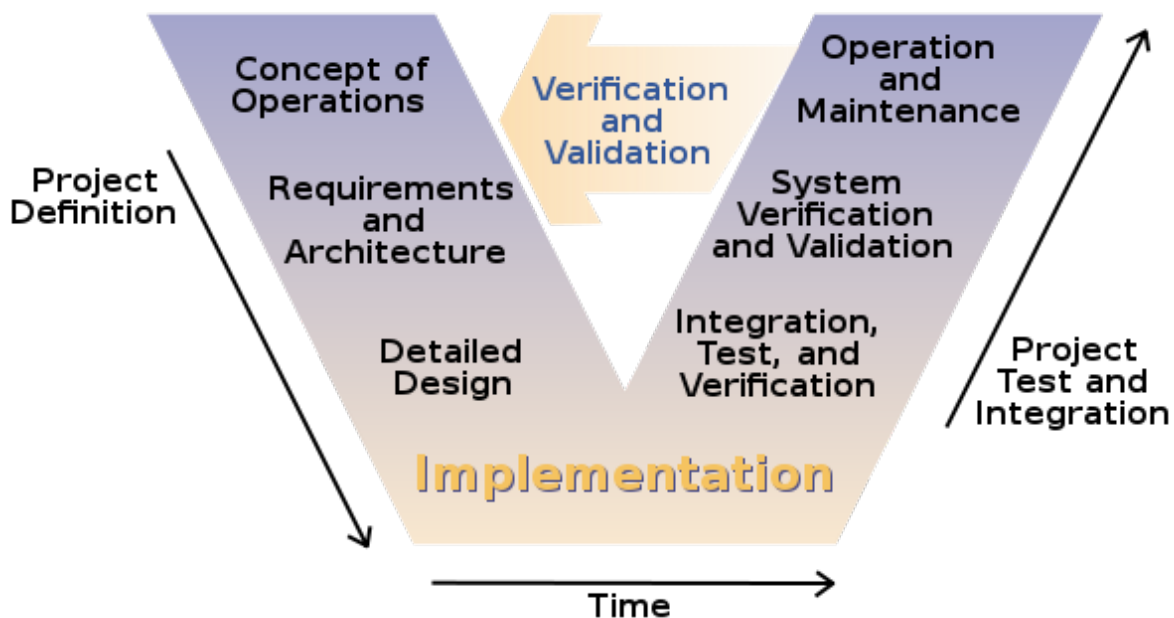
**TERM TO KNOW****Software Development Methodology**

Various attempts to split the process of software development into separate phases.

2. Cascade/V-Model Development

The cascading methodology is a linear method in which development occurs sequentially through phases. The cascading methodology is commonly referred to as the waterfall method. In this method, the project is broken down into phases that occasionally overlap to ensure consistency across the entire project. The steps at the beginning of the method flow down into subsequent steps, similar to a waterfall. In this methodology, all progress must move in one direction with heavy emphasis placed on the planning, delivery dates, and implementation of the software developed.

The V-Model approach to software development maintains phases that are very similar to the cascading or waterfall model. With this methodology, the process steps move down as in the cascading method, but are then bent upwards after the design phase, making a "V" shape.



The cascading or V-model was one of the first software development methodologies introduced, and thus has its limitations, which led to the development of other models. In particular, this model limits efficiencies regarding requirement change management. This limitation actually motivated the development of lean and agile approaches, discussed in more detail below.

IN CONTEXT

Bison Software Game Creator Project

In our Game Creator project example, if management were to utilize a cascading/V-model methodology, the Game Creator project may play out as follows:

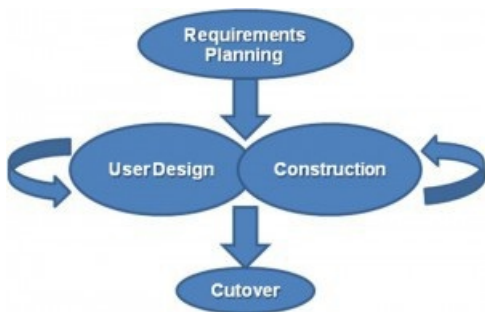
- **Concept of Operations:** During this phase, a review of the Game Creator idea is completed. Questions such as the following are asked: Is creating a solution possible? What alternatives exist? What is currently being done about it? Is this project a good fit for our organization? This phase will ultimately determine if Game Creator will even get started.
- **Requirements and Architecture:** In this phase, the system analyst will work with different stakeholder groups to determine the specific requirements for Game Creator. No programming is done in this phase. Instead, procedures are documented, key players are interviewed, and data requirements are developed in order to get an overall picture of exactly what the Game Creator is supposed to do. The result of this phase is a system-requirements document.
- **Detailed Design:** In this phase, the designer takes the system-requirements document created in the previous phase, and develops the specific technical details required for Game Creator. It is in this phase that the business requirements are translated into specific technical requirements. The designs for Game Creator's user interface, database, data inputs and outputs, and reporting are developed here. The result of this phase is a system-design document. This document will have everything a programmer will need to actually create the system.
- **Programming:** Game Creator code finally gets written in the programming phase. Using the system-design document as a guide, a programmer (or team of programmers) develops the

program. The result of this phase is an initial working program that meets the requirements laid out in the system-analysis phase and the design developed in the system-design phase.

- **Integration, Test, and Verification:** In the testing phase, the Game Creator version developed in the previous phase is put through a series of structured tests. The first is a unit test, which tests individual parts of the code for errors or bugs. Next is a system test, where the different components of the system are tested to ensure that they work together properly. Finally, the user-acceptance test allows those who will be using the software a chance to test the system to ensure that it meets their standards. Any bugs, errors, or problems found during testing are addressed and then tested again.
- **System Verification and Validation:** Implementation. Once Game Creator is developed and tested, it has to be implemented in the organization. This phase includes training the users, providing documentation, and conversion from any previous system to the new version of Game Creator. Implementation can take many forms, depending on the type of system, the number and type of users, and how urgent it is that the system become operational.
- **Operation and Maintenance:** This final phase takes place once the implementation phase is complete. In this phase, the system has a structured support process in place: reported bugs are fixed and requests for new features are evaluated and implemented; and system updates and backups are performed on a regular basis.

3. Rapid Application Development (RAD)

Rapid application development (RAD) is a software-development (or systems-development) methodology that focuses on quickly building a working model of the software, getting feedback from users, and then using that feedback to update the working model. After several iterations of development, a final version is developed and implemented.



IN CONTEXT

Bison Software Game Creator Project

In our Game Creator project example, if management were to decide on using RAD to develop Game Creator, it might play out as follows:

- **Requirements Planning:** In this RAD phase, the Bison Software team's project manager, testers, and analysts would meet to discuss the resources the Game Creator project will need, the project's scope, system requirements, project language, and project deadlines. This phase

would not end until all stakeholders agree on all relevant issues related to the Game Creator project.

- **User Design:** In this RAD phase, the Bison Software team may have its testers, quality assurance personnel, users from the general public, or combinations of the three groups work with the team's system analyst to discuss specific user needs for the Game Creator application. The analyst will then translate these user needs to programmers, or software developers, to develop a working prototype that can be evaluated by potential users. The systems analyst will collect feedback related to the prototype.
- **Construction:** In this RAD phase, the Bison Software team now has its programmers and software developers writing the software code in the agreed-upon software language. The programmers are generally developing the application based on the prototype and the user feedback suggestions provided. Users may still be allowed to participate in this phase to provide feedback or suggestions as deliverables are created (i.e. phone).
- **Cutover:** In this RAD phase, the Bison Software team are ready to implement the Game Creator application. However, more testing is still done and feedback provided to the systems analyst. After the application has been created and all parties are satisfied, Game Creator can be released.

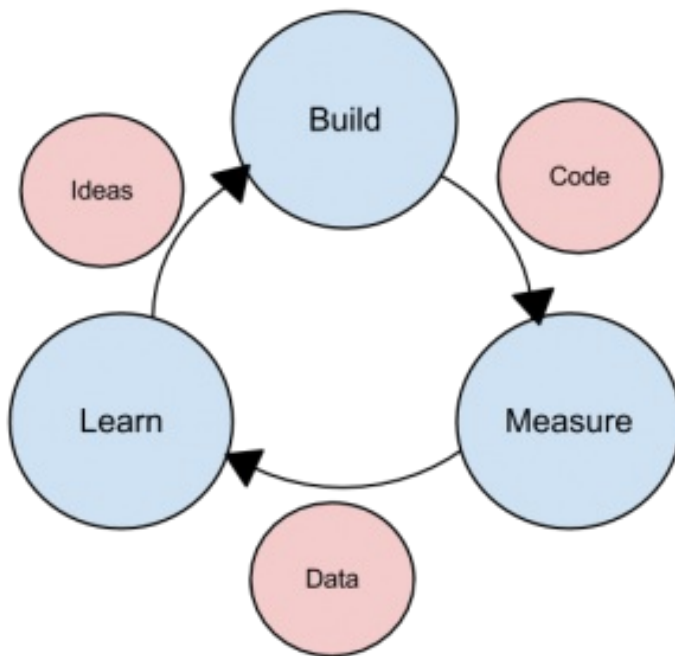
4. Agile Development

Agile methodologies are a group of methodologies that utilize incremental changes with a focus on quality and attention to detail. Each increment is released in a specified period of time (called a time box), creating a regular release schedule with very specific objectives. The goal of the agile methodologies is to provide the flexibility of an iterative approach while ensuring a quality product. The characteristics of agile methods include:

- Small cross-functional teams that include development-team members and users;
- Daily status meetings to discuss the current state of the project;
- Short time-frame increments (from days to one or two weeks) for each change to be completed; and
- At the end of each iteration, a working project is completed to demonstrate to the stakeholders.

5. Lean Development

The Lean methodology focuses on taking an initial idea and developing a minimum viable product (MVP). The minimum viable product is a product that has been developed with the sole intent of being further developed in future phases of a project. The MVP has just enough functionality to provide the development team with feedback on how to improve the application. Once the MVP is developed, it is given to potential users for review. Feedback on the MVP is generated in two forms: (1) direct observation and discussion with the users, and (2) usage statistics gathered from the software itself. Using these two forms of feedback, the team determines whether it should continue in the same direction or rethink the core idea behind the project, change the functions, and create a new MVP. This change in strategy is called a pivot. Several iterations of the MVP are developed, with new functions added each time based on the feedback, until a final product is completed.



IN CONTEXT

Bison Software Game Creator Project

In our Game Creator project example, a lean development methodology would be employed by management if the initial interest is to start with a functional version of Game Creator that, based on test data, could be continually developed until a final product is completed. Therefore, as soon as Bison Software has the idea for Game Creator, its development is commissioned, and a development team is tasked with the project. Based on the lean methodology image above, the Game Creator project might play out as follows:

- **Idea:** Based on game player data from the Bison Software website, management thinks that it would be a good idea to develop a software application that will provide users with the ability to create games on their iPhones or iPads. Management decides to call the application “Game Creator” and immediately assigns a project manager who, subsequently, puts together a development team to work on the Game Creator project. At this point, the development team has no software requirements. The team has only been given an idea and tasked with building an MVP. This is a very big difference from the other development methodologies.
- **Build:** During this phase, the development team collaborates to choose a software language and develop a minimum viable product MVP. This phase ends when the MVP is developed.
- **Measure:** Once the MVP is developed, its viability is immediately measured. This phase is completed by giving the MVP to potential users for review. Feedback on the MVP is generated in two forms: (1) direct observation and discussion with the users, and (2) usage statistics gathered from the software itself.
- **Data:** During this phase, the data is recorded and visualizations are generated, based on the data acquired during the measurement phase. In this example, the data might show user trends, preferences, game play mechanics, process analysis, etc. Data points, sets, and groupings are generated and put into the Bison Software information system and data warehouse.

- **Learn:** At this phase of development, the data is analyzed and discussed by all Game Creator key stakeholders. Based on the conclusions arrived during these meetings, Game Creator requirements are generated. Based on these new requirements and the opinions of management, an ultimate determination is made as to whether they should continue in the same direction or rethink the core idea behind the project, change the functions, and create a new MVP.

If management deems it necessary to continue, the Game Creator project will continue by restarting at the idea phase, but during this run the team will have the software requirements and an MVP. The MVP will be further refined by going through the same process.

6. Prototyping

The prototyping software development methodology seeks to develop multiple prototypes of the software program being developed. Most developers do not view this as a complete methodology. In fact, many developers view prototyping as a phase of a complete methodology such as RAD. The main goal of prototyping is to develop a working system that can then be built upon until a full system, with all project requirements, is completed.

IN CONTEXT

Bison Software Game Creator Project

In our Game Creator Project example, if the prototyping methodology is to be employed, management may deploy several software development teams to each develop a prototype of the Game Creator application. Potential users, in addition to Bison managers, would be assigned to each team and heavily involved in the process.



SUMMARY

Software development is about so much more than programming. Developing new software applications requires several steps, from the formal SDLC process, to more informal processes such as '**rapid application development, agile programming, or lean methodologies**'. More and more industry development teams are utilizing several **development methodologies** on one project. In many cases, software developers create a hybrid of these methodologies.

Source: Derived from Chapter 10 of "Information Systems for Business and Beyond" by David T. Bourgeois. Some sections removed for brevity.

<https://www.saylor.org/site/textbooks/Information%20Systems%20for%20Business%20and%20Beyond/Textbook.html>, Image of V-Model, Public Domain [https://en.wikipedia.org/wiki/V-Model_\(software_development\)/#/media/File:Systems_Engineering_Process_II.svg](https://en.wikipedia.org/wiki/V-Model_(software_development)/#/media/File:Systems_Engineering_Process_II.svg)



TERMS TO KNOW

Software Development Methodology

Various attempts to split the process of software development into separate phases.