

March 15, 2012

Software Development Life Cycle Models and Methodologies

Posted by Mohamed Sami · Requirements Engineering, Software Engineering

320 Votes

Introduction

Software development life cycle (**SDLC**) is a series of phases that provide a common understanding of the software building process. How the software will be realized and

developed from the business understanding and requirements elicitation phase to convert these business ideas and requirements into functions and features until its usage and operation to achieve the business needs. A good software engineer should have enough knowledge on how to choose the SDLC model based on the project context and the business requirements.

Therefore, it may be required to choose the right SDLC model according to the specific concerns and requirements of the project to ensure its success. I wrote another article on how to choose the right SDLC, you can follow this [link](#) for more information. Moreover, to learn more about [Software Testing life cycles](#) and [SDLC phases](#) you follow the links highlighted here.

In this article, we will explore the different types of SDLC models and the advantages and disadvantages of each one, and when to use them.

You can think of SDLC models as tools that you can use to better deliver your software project. Therefore, knowing and understanding each model and when to use it, the advantages and disadvantages of each one are important to know which one is suitable for the project context.



Tools

Types of Software developing life cycles (SDLC)

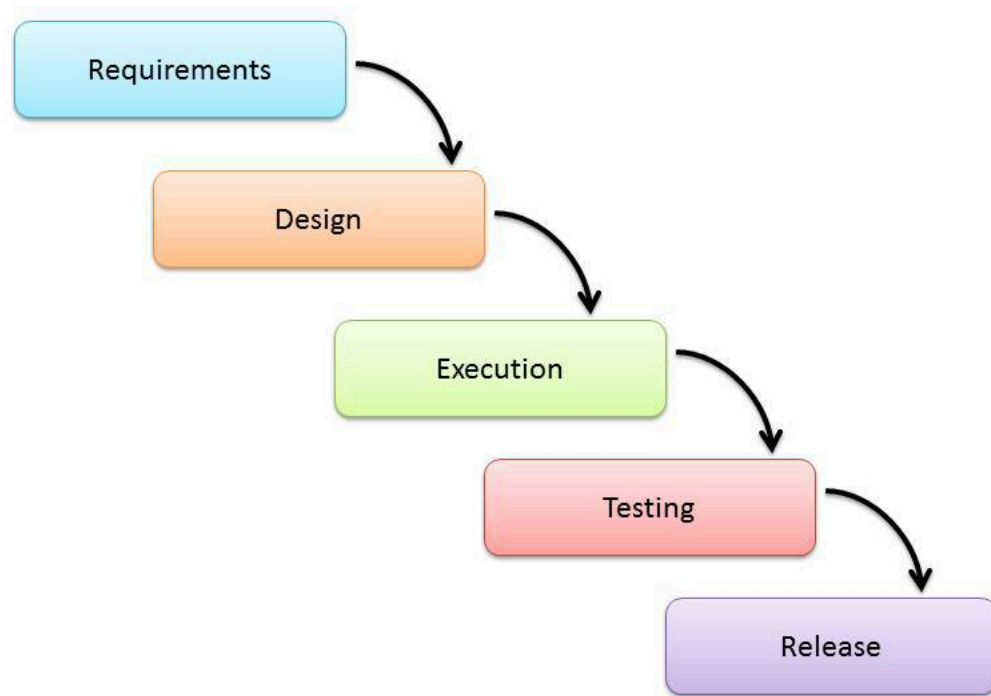
- [Waterfall Model](#)
- [V-Shaped Model](#)
- [Evolutionary Prototyping Model](#)

- Spiral Method (SDM)
- Iterative and Incremental Method
- Agile development

Waterfall Model

Description

The Waterfall Model is a linear sequential flow. In which progress is seen as flowing steadily downwards (like a waterfall) through the phases of software implementation. This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement. The waterfall approach is the earliest approach and most widely known that was used for software development.



Waterfall Model

The usage

For projects which not focus on changing the requirements, for example, projects initiated from a request for proposals (RFPs), the customer has very clear documented requirements

Advantages and Disadvantages

| | |
|------------|---------------|
| Advantages | Disadvantages |
|------------|---------------|

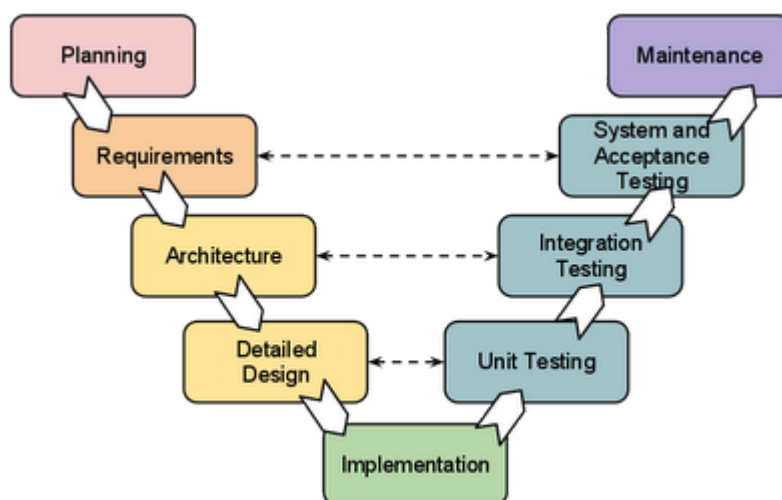
| | |
|---|---|
| <ul style="list-style-type: none"> • Easy to explain to the users. • Structures approach. • Stages and activities are well defined. • Helps to plan and schedule the project. • Verification at each stage ensures early detection of errors/misunderstandings. • Each phase has specific deliverables. | <ul style="list-style-type: none"> • Assumes that the requirements of a system can be frozen. • Very difficult to go back to any stage after it is finished. • A little flexibility and adjusting the scope are difficult and expensive. • Costly and required more time, in addition to the detailed plan. |
|---|---|

Advantages/Disadvantages table

V-Shaped Model

Description

It is an extension of the waterfall model, Instead of moving down in a linear way, the process steps are bent upwards after the implementation and coding phase, to form the typical V shape. The major difference between the V-shaped model and the waterfall model is the early test planning in the V-shaped model.



V-Shaped Model

The usage

- Software requirements clearly defined and known
- Software development technologies and tools are well-known

Advantages and Disadvantages

| Advantages | Disadvantages |
|---|---|
| <ul style="list-style-type: none">• Simple and easy to use• Each phase has specific deliverables.• Higher chance of success over the waterfall model due to the development of test plans early on during the life cycle.• Works well for where requirements are easily understood.• Verification and validation of the product in the early stages of product development. | <ul style="list-style-type: none">• Very inflexible, like the waterfall model.• Adjusting the scope is difficult and expensive. The software is developed during the implementation phase, so no early prototypes of the software are produced. The model doesn't provide a clear path for problems found during the testing phases.• Costly and required more time, in addition to a detailed plan |

Advantages/Disadvantages table

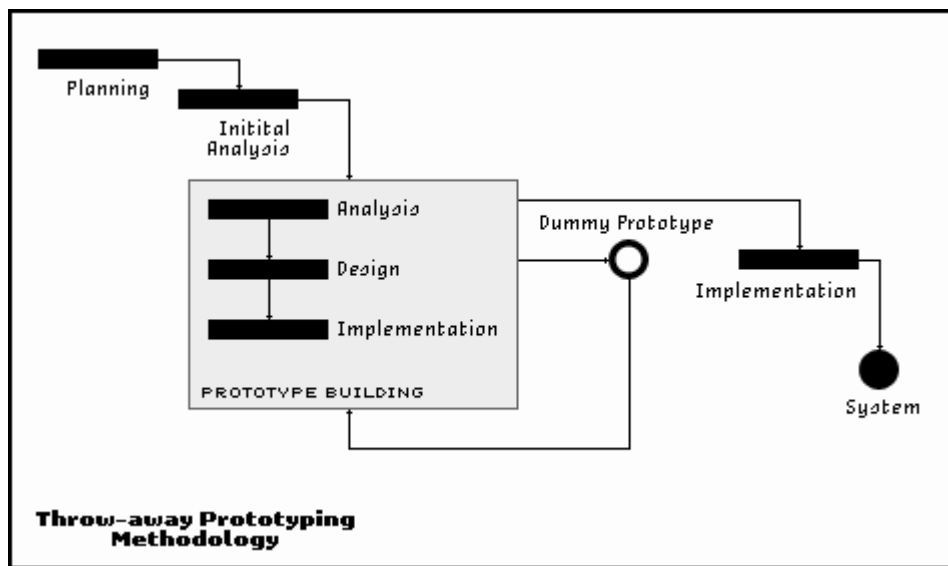
Prototyping Model

Description

It refers to the activity of creating prototypes of software applications, for example, incomplete versions of the software program being developed. It is an activity that can occur in software development and It used to visualize some components of the software to limit the gap of misunderstanding the customer requirements by the development team. This also will reduce the iterations that may occur in the waterfall approach and are hard to be implemented due to the inflexibility of the waterfall approach. So, when the final prototype is developed, the requirement is considered to be frozen.

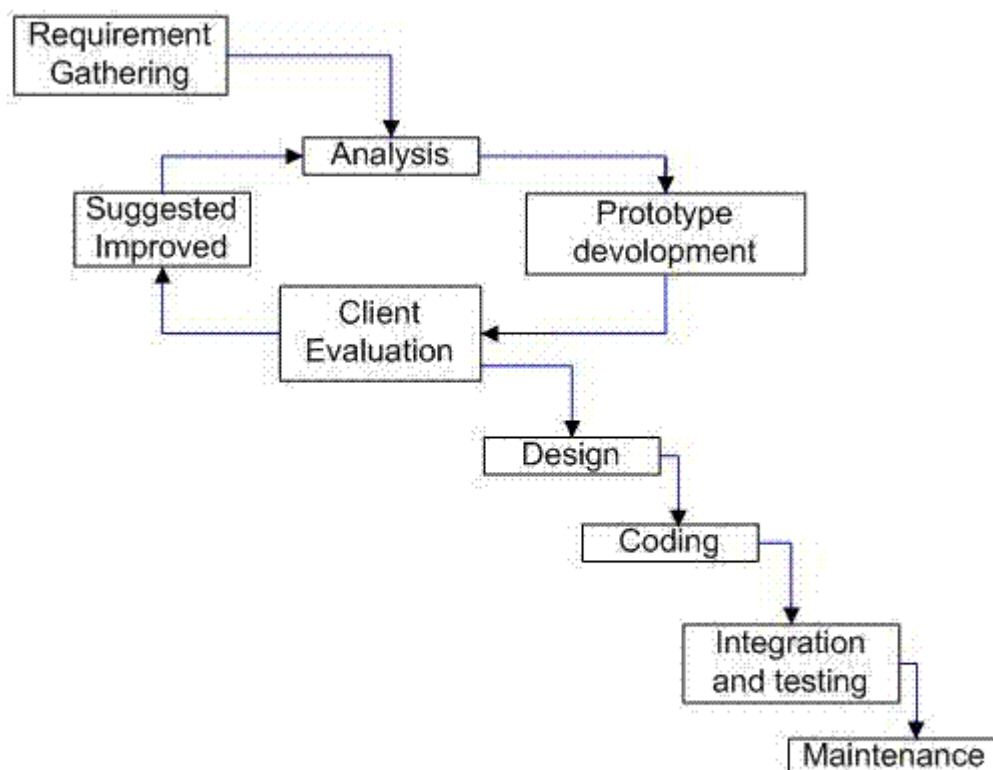
It has some types, such as:

- Throwaway prototyping: Prototypes that are eventually discarded rather than becoming a part of the finally delivered software



Throwaway prototyping

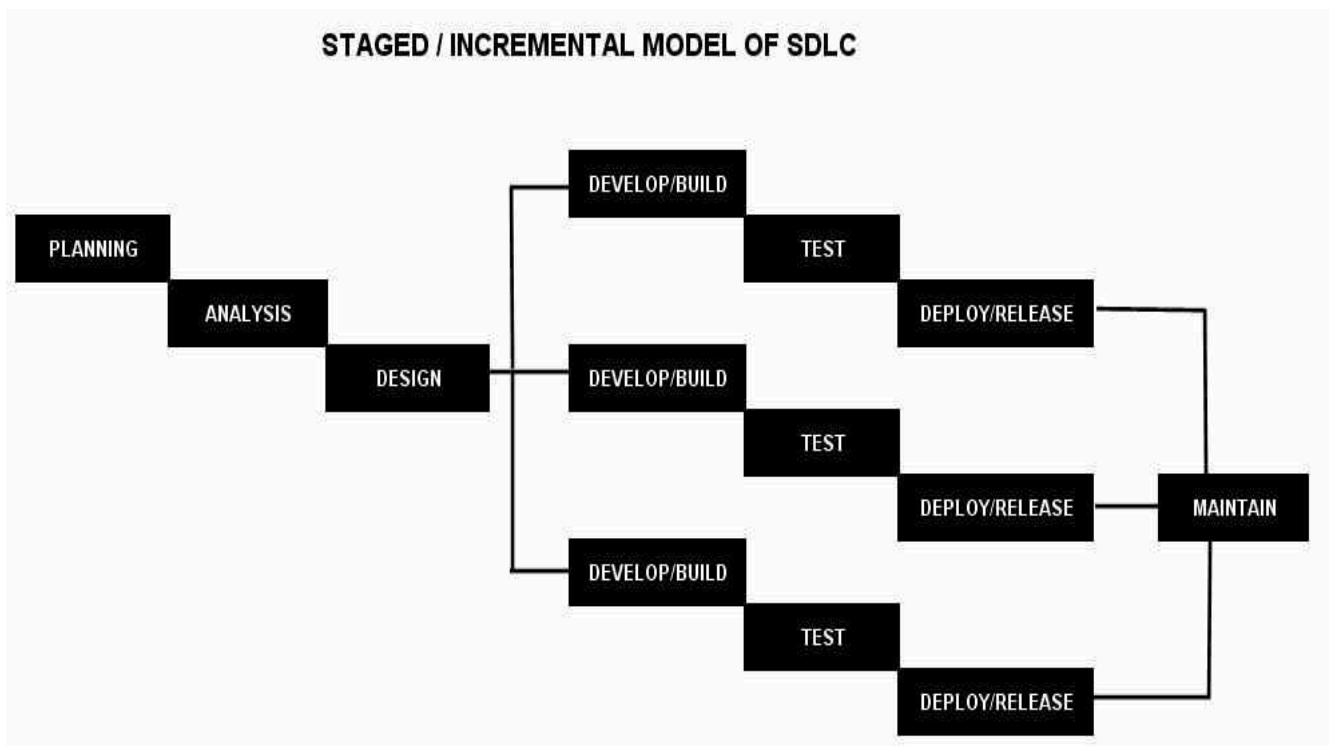
- Evolutionary prototyping: prototypes that evolve into the final system through an iterative incorporation of user feedback.



Evolutionary Prototyping Model

Evolutionary prototyping

- Incremental prototyping: The final product is built as separate prototypes. In the end, the separate prototypes are merged in an overall design.



Incremental prototyping

- Extreme prototyping: used in web applications mainly. Basically, it breaks down web development into three phases, each one based on the preceding one. The first phase is a static prototype that consists mainly of HTML pages. In the second phase, the screens are programmed and fully functional using a simulated services layer. In the third phase, the services are implemented

The usage

- This process can be used with any software developing life cycle model. While this shall be chosen when you are developing a system has user interactions. So, if the system does not have user interactions, such as a system does some calculations shall not have prototypes.

Advantages and Disadvantages

| Advantages | Disadvantages |
|--|---|
| <ul style="list-style-type: none"> • Reduced time and costs, but this can be a disadvantage if the developer loses time in developing the prototypes. • Improved and increased user involvement. | <ul style="list-style-type: none"> • Insufficient analysis. User confusion of prototype and finished system. • Developer misunderstanding of user objectives. |

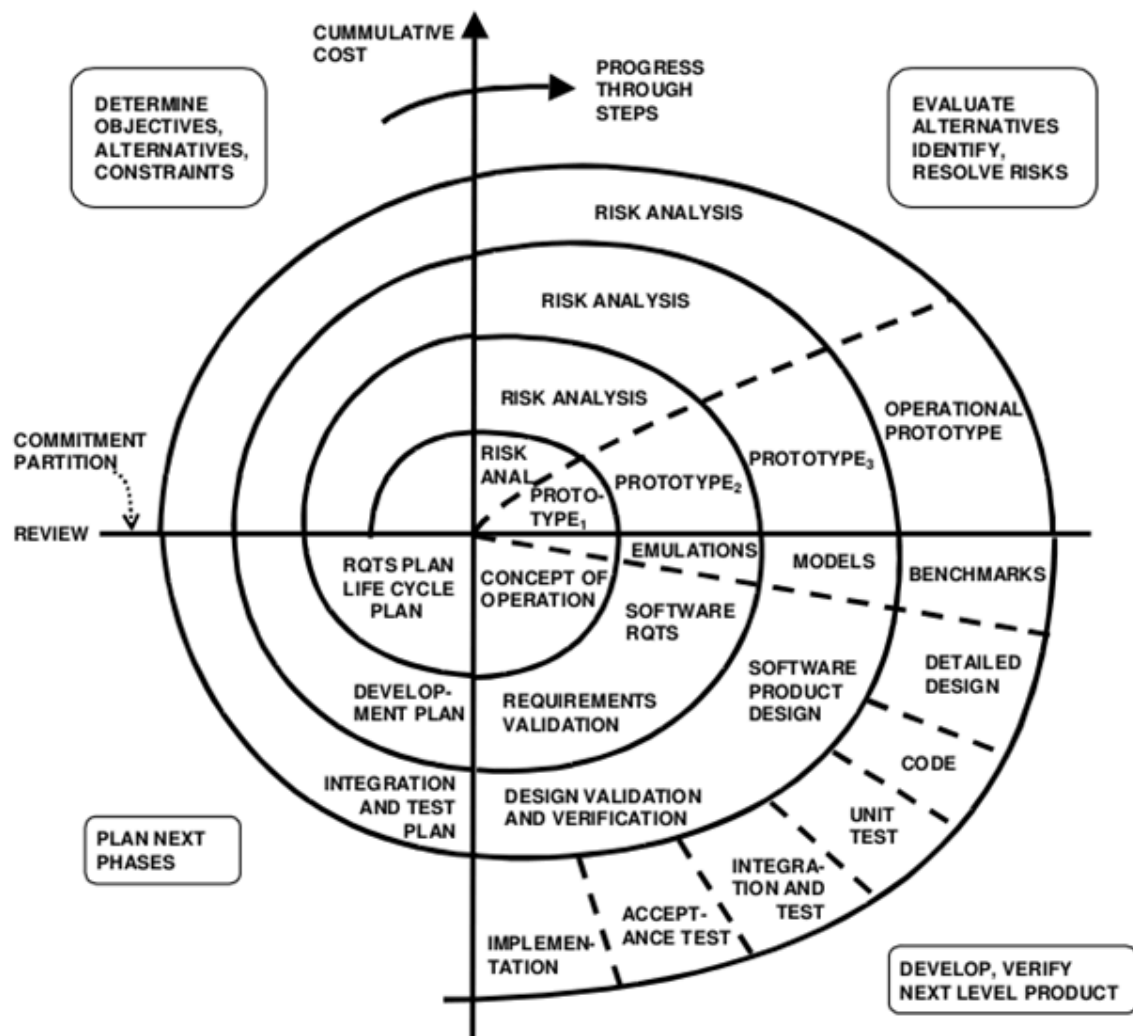
- | | |
|--|---|
| | <ul style="list-style-type: none">• Excessive development time of the prototype.• It is costly to implement the prototypes |
|--|---|

Advantages/Disadvantages table

Spiral Model (SDM)

Description

It is combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. This model of development combines the features of the prototyping model and the waterfall model. The spiral model is favored for large, expensive, and complicated projects. This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations.



Spiral Model

The usage

It is used in large applications and systems which built-in small phases or segments.

Advantages and Disadvantages

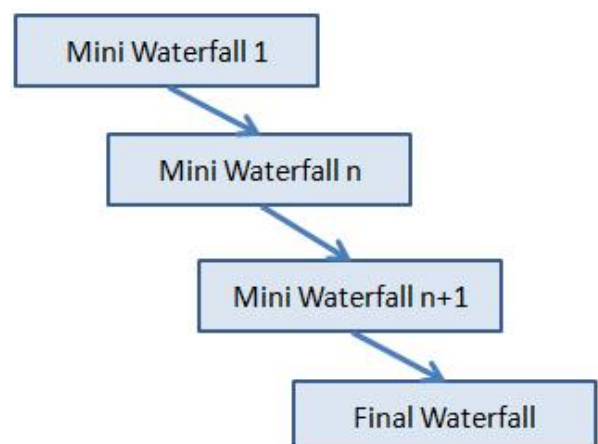
| Advantages | Disadvantages |
|---|--|
| <ul style="list-style-type: none">• Estimates (i.e. budget, schedule, etc.) become more realistic as work progressed because important issues are discovered earlier.• Early involvement of developers.• Manages risks and develops the system into phases. | <ul style="list-style-type: none">• High cost and time to reach the final product.• Needs special skills to evaluate the risks and assumptions.• Highly customized limiting re-usability |

Advantages/Disadvantages table

Iterative and Incremental Model

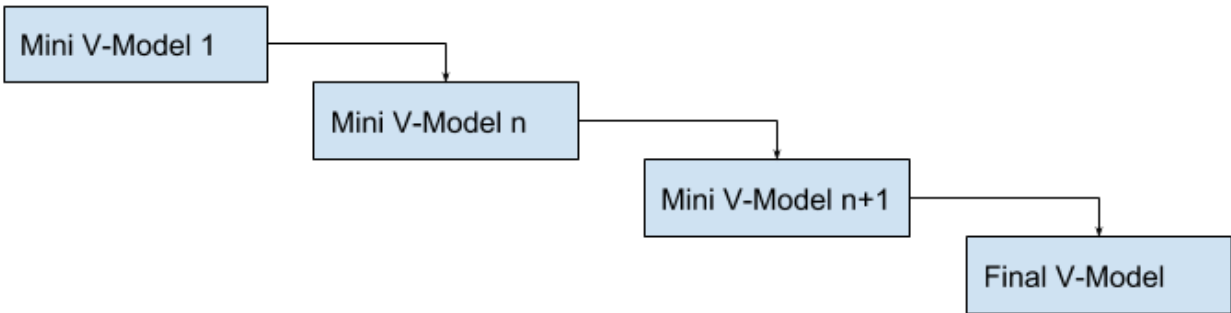
Description

It is developed to overcome the weaknesses of the waterfall model. It starts with initial planning and ends with deployment with the cyclic interactions in between. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during the development of earlier parts or versions of the system. It can consist of mini waterfalls or mini V-Shaped models.



The usage

It is used in shrink-wrap applications and large systems which built-in small phases or segments. Also, can be used in a system that has separated components, for example, an ERP system. Which we can start with the budget module as a first iteration and then we can start with the inventory module and so forth.



Iterative and Incremental Model

Advantages and Disadvantages

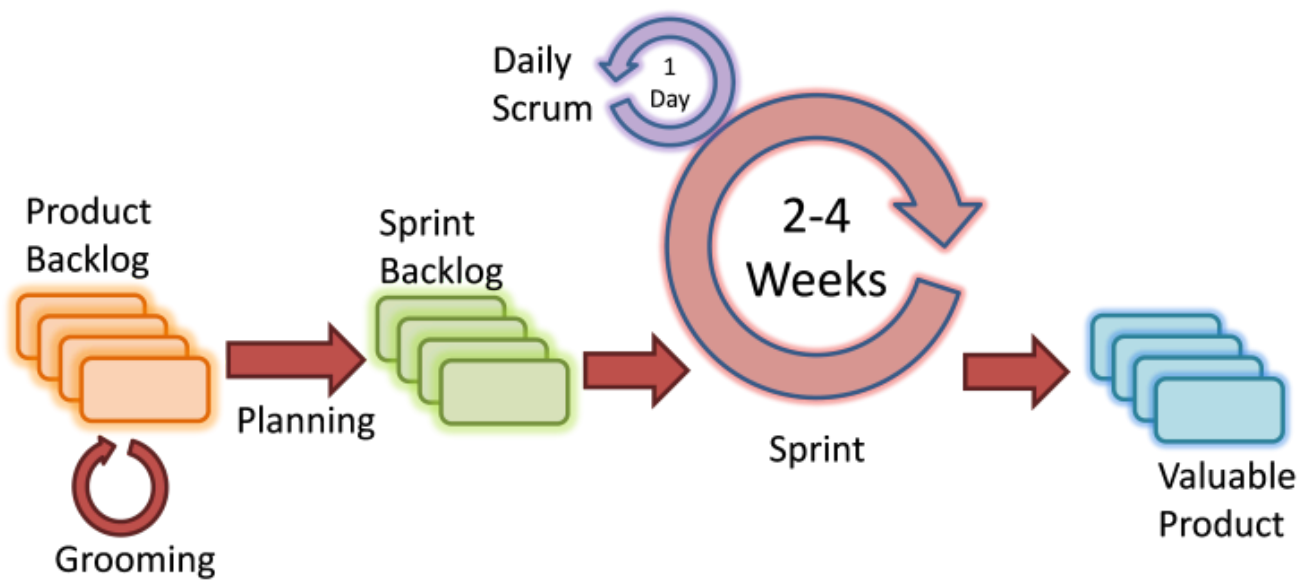
| Advantages | Disadvantages |
|---|---|
| <ul style="list-style-type: none">• Produces business value early in the development lifecycle.• Better use of scarce resources through proper increment definition.• Can accommodate some change requests between increments.• More focused on customer value than the linear approaches.• We can detect project issues and changes earlier. | <ul style="list-style-type: none">• Requires heavy documentation.• Follows a defined set of processes.• Defines increments based on function and feature dependencies.• Requires more customer involvement than the linear approaches.• Partitioning the functions and features might be problematic.• Integration between the iterations can be an issue if it is not considered during the development and project planning. |

Advantages/Disadvantages table

Agile Model

Description

It is based on iterative and incremental development, where requirements and solutions evolve through collaboration between cross-functional teams.



Scrum Agile Model

The usage

It can be used with any type of project, but it needs more engagement from the customer and to be interactive. Also, we can use it when the customer needs to have some functional requirements ready in less than three weeks and the requirements are not clear enough. This will enable more valuable and workable pieces for software early which also increases customer satisfaction.

Advantages and Disadvantages

| Advantages | Disadvantages |
|---|---|
| <ul style="list-style-type: none"> • Decrease the time required to avail some system features. • Face-to-face communication and continuous inputs from customer representatives leave no space for guesswork. The end result is high-quality software in the least possible time duration and satisfied customers | <ul style="list-style-type: none"> • Scalability. • The ability and collaboration of the customer to express user needs. • Documentation is done at later stages. • Reduce the usability of components. • Needs special skills for the team. |

Advantages/Disadvantages table

Here is a consolidated presentation to illustrate most of the popular software development models.

Saltar a la siguiente diapositiva Podrás ir a la siguiente diapositiva en 3

Anuncio

Saltar a la siguiente diapositiva Podrás ir a la siguiente diapositiva en 3

Anuncio

Saltar a la siguiente diapositiva Podrás ir a la siguiente diapositiva en 3

Anuncio

[cite]

DONATE TODAY

Help to do more!

The content you read is available for free. If you've liked any of the articles at this site, please take a second to help us write more and more articles based on real experiences and maintain them for you and others. Your support will make it possible for us.

\$10.00

Also published on [Medium](#).

Summary



Article Name

Software Development Life Cycle Models and Methodologies

Description

Software development life cycle (SDLC) is a series of phases that provide a common understanding of the software building process.

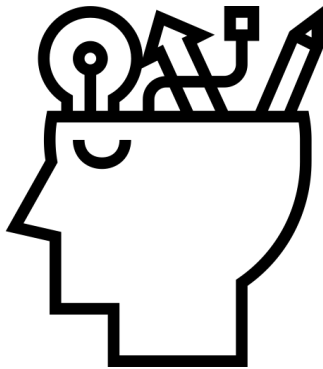
Author

Mohamed Sami

Publisher Name

<https://melsatar.blog>

Publisher Logo



394 responses to “Software Development Life Cycle Models and Methodologies”

The Different Software Processes And How They Impact Software Architecture – Housing Prototypes

December 9, 2022

[...] these predictions' averages, which reduces the likelihood of them occurring. The selection of software lifecycle models that are based on desired process qualities has a direct impact on how phases are carried out and [...]

Loading...

0 0 Rate This

[Reply](#)

[Der geschäftliche Wert des kontinuierlichen Testens • Wons](#)
[September 28, 2022](#)

[...] um Geschäftsziele in einem organisatorischen Setup zu rationalisieren. Üben und umsetzen Softwareentwicklungsmodelle ist sowohl eine Geschäftsentscheidung als auch eine IT-Initiative. IT-Abteilungen können [...]

Loading...

0 0 Rate This

[Reply](#)

[El valor comercial de las pruebas continuas • ARUF](#)
[September 12, 2022](#)

[...] los objetivos comerciales en una configuración organizacional. Practicando e implementando modelos de desarrollo de software es tanto una decisión comercial como una iniciativa de TI. Los departamentos de TI pueden [...]

Loading...

0 0 Rate This

[Reply](#)

[Коммерческая ценность непрерывного тестирования • ARUF](#)
[September 5, 2022](#)

[...] в организационной структуре. Практика и реализация модели разработки программного обеспечения является как деловым решением, так и ИТ-инициативой. [...]

Loading...

0 0 Rate This

[Reply](#)

[Older Comments](#)

[1](#) ... [9](#) [10](#) [11](#)

Let me know your thoughts

About the author

Mohamed Sami is a Industry Advisor who has a solid engineering background, he has more than 18 years of professional experience and he was involved in more than 40 government national projects with holding different roles and responsibilities, from national projects execution and management to drafting of the conceptual architecture and solutions design. Furthermore, Mohamed contributed to various digital strategies in the government sector, which improved his business and technical skills over his career development.

Related Posts

Software Development Life Cycle (SDLC)

March 14, 2016

Choosing the right Software development life cycle model

March 21, 2012

The Waterfall Model, a different perspective

February 16, 2018

Mohamed Sami

Personal blog – Digital World, Software Engineering & Architecture Practices

Get In Touch

Search

[Home](#)

[Articles](#)

[Contact me](#)

[About](#)

[Languages](#)

All rights reserved ©

Share