

Examine the case for
the waterfall life
cycle. does the
waterfall life cycle
stil...



**ASSIGN
BUSTER**

Examine the case for the waterfall life cycle. Does the waterfall life cycle still play a useful role in modern system development, and if so, at which stages?

The waterfall life cycle model in software engineering was originally designed in 1970 by Winston. W. Royce. The waterfall life cycle model plays a linear act of moving forward of framework activities that is often inconsistent in some field of the software world. In situations where requirements are well defined and stable, it is applicable to play a vital role.

The waterfall model is easy to understand and reinforces the notion of "define before design" and "design before code". The model looks for the complete requirements at the beginning of the process which is unrealistic.

The waterfall model is one of the traditional or classic models used in the software development field which is very easy to use and simple to understand. The name waterfall is suggested due to the similarity of the model with the cascade of waterfalls. In the waterfall life cycle, the whole process is separated into independent phases. All the phases are cascade to one another so that the output of the present phase would be the input for the next phase. All the methods and processes attempted in the model are more visible. The highlight of the waterfall model is that, there are different phases in the model life cycle and the existing phase must be finished before the next phase starts. The five phases of the waterfall model are bulleted below in sequence: 1) Requirements Definition 2) System and Software Design 3) Implementation and Unit testing 4) Integration and System testing 5) Operation and Maintenance Requirement Definition: All the requirements that are expected by the end user from the system are collected in this phase. The aim of this phase is to estimate the time and cost needed for the

<https://assignbuster.com/examine-the-case-for-the-waterfall-life-cycle-does-the-waterfall-life-cycle-still-play-a-useful-role-in-modern-system-development-and-if-so-at-which-stages/>

development of the system. The possible risks are analyzed and studied then a document is prepared which will carry the listed requirement for the next phase. System and Software Design: The phase emphasizes on how should the end product looks like before writing a single line of code. A system design is prepared in this phase after studying the requirement specifications from the previous phase that helps to create a design document which serve as an input for the next phase. Implementation and Unit testing: The actual coding begins in this phase after receiving the design document from the previous phase. The system is implemented into a small program known as a unit and these units are tested for functionality which is referred as unit testing. Integration and System testing: The units are integrated into a complete system in this phase. The complete system is tested as a whole to check if the units are coordinate with each other and the whole system is behaving as per user specifications or not. Finally, the software is delivered to the end user after successfully testing it. Operation and Maintenance: After delivering the software with a successful testing, many problems remain in the system which arises time to time when it is practically used by the user. These problems are rectified and solved in the maintenance phase. Figure: Waterfall Model (Royce, 1970) In which stage the waterfall model plays a useful role? There is a mild state of dissociation of work and command in the waterfall model. For this it improves its qualities by ensuring minimal wastage of time and effort before starting the actual coding which reduces the risk of schedule, slippage or customer expectation and this way it become easier to catch and correct the faults at the design stage then at the testing stage. Finally, the first two phases terminated in

<https://assignbuster.com/examine-the-case-for-the-waterfall-life-cycle-does-the-waterfall-life-cycle-still-play-a-useful-role-in-modern-system-development-and-if-so-at-which-stages/>

the production of a formal specification so the waterfall cycle can aid effective knowledge transfer among the team members at distributed locations. Some of the limitations imposed by waterfall model The waterfall model has some noticeable limitations like the one has to finalize the requirements before the project commences. The redesigning of physical system is difficult when a phase is complete as it is closed and cannot be revisited. The waterfall model fails to handle the different types of risks in a project that might occur at a later stage of the project which results in the badly structured system. The waterfall model is strictly sequential as well as inadequate analysis can be noticed only at the end of the development. In waterfall model scalability and complexity are generally based on how application system being developed. In waterfall model flaws are detected mainly in implementation and system testing phases. End users are not involved in the development stage so it is the responsibility of the analysts and programmers to understand the end user requirements which is a quite risky process. To achieve a realistic and a better efficient system, most real-life projects cannot follow the rigid sequence imposed by the waterfall life cycle model. How iterative model plays a useful role rather than waterfall model in modern system development? The cyclic repetition of tasks is termed as iteration. The Iterative life cycle is a common thread in many current approaches to software development. In iterative development, revisions of work on the earlier phases of the system can be made. The iterative model is made up of a series of activities that can be repeated. Iterations deals mainly with the most important risks first. Iterative and incremental model is a step by step approach which was introduced by Berry

<https://assignbuster.com/examine-the-case-for-the-waterfall-life-cycle-does-the-waterfall-life-cycle-still-play-a-useful-role-in-modern-system-development-and-if-so-at-which-stages/>

Boehm in the year 1988. Figure: Iterative or Spiral model. The diagrammatic representation of these models appears like a spiral with many loops so it is also known as the spiral model. Each phase in this model is divided into four quadrants. The first quadrant identifies risk analysis based on initial requirements. The second quadrant helps in initial requirements and project planning. The third quadrant is based on the customer evaluation. At last the fourth quadrant is concerned with the engineering with reviewing the results of the stages traversed a customer and planning the next loop around the spiral. At the inception, risks and errors are identified and resolve for developing an appropriate prototype. At the second phase, planning is carried out. A strict risk judgment is performed. The development process builds the prototype and it is made available to the customer for evaluation. As a drawback of the model the following point needs to be attended, like the process guidance towards objective determination. Risk assessment depends on the experience of the development team not on the model by itself. In conclusion, although there are few disadvantages of the waterfall model it remains to be one of the popular software development approach in the industry. The waterfall approach is more suited to projects where requirements are stable and not likely to change or where there is less need for user involvement. However, with increasing features to successful development many development projects have moved away from the waterfall life cycle. As opposed to the waterfall model the iterative and the incremental development can be viewed as a goal oriented model because it is a step by step process stresses on the risk factor whereas waterfall model describe the steps. As the complete model cannot be viewed at a time and

<https://assignbuster.com/examine-the-case-for-the-waterfall-life-cycle-does-the-waterfall-life-cycle-still-play-a-useful-role-in-modern-system-development-and-if-so-at-which-stages/>

the changes to the process occurs with the changing scenario. The spiral model is capable of handling such situations. Finally, if the team does not have a lot of experience, the waterfall model perfectly serves the purpose.

References Bennett, S. et al. (2010) Meeting the Challenges. In: Hoyle, C. et al. (eds.) Object-Oriented Systems Analysis and Design. 4th ed. Berkshire: McGraw-Hill Education, pp. 71-77. Waterfall model explained (2011) [Online]. Available from: <http://www.buzzle.com/editorials/1-5-2005-63768.asp>. [Accessed 6/1/13]. Self-Assessments for OOP Introduction (2006) Spiral model [Online image]. Available from: <http://elm.eeng.dcu.ie/~ee553/ee402notes/html/ch01s05.html>. [Accessed 5/12/12]. Royce (1970) Waterfall model [Online image]. Available from: <http://www.robabdul.com/Data-Management-System-Software-Development-Cycle.asp>. [Accessed 5/12/12]. Aggarwal and Singh (2007) Software life cycle model. 3rd ed. India: New age international publisher. Kuchta, J. (2010). Software Life Cycle, available from: http://lib.bioinfo.pl/files/courses/pdfcache/lecture_229.pdf. [Accessed 6/1/13]. Learnaccessvba. (2006). Waterfall Development Methodology, available from: http://www.learnaccessvba.com/application_development/waterfall_method.htm [Accessed 6/1/13]. Sommerville, I.: (2004) Software Engineering, 7th ed. Pearson Education Ltd., London

<https://assignbuster.com/examine-the-case-for-the-waterfall-life-cycle-does-the-waterfall-life-cycle-still-play-a-useful-role-in-modern-system-development-and-if-so-at-which-stages/>