**SDLC**

The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process. SDLC is a structure followed by a development team within the software organization. It consists of a detailed plan describing how to develop, maintain and replace specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.   
This term is also known as the software development process.

SDLC consists of following activities:

1. Planning: The most important parts of software development, requirement gathering, or requirement analysis are usually done by the most skilled and experienced software engineers in the organization. After the requirements are gathered from the client, a scope document is created in which the scope of the project is determined and documented.
2. Implementation: The software engineers start writing the code according to the client's requirements.
3. Testing: This is the process of finding defects or bugs in the created software.
4. Documentation: Every step in the project is documented for future reference and for the improvement of the software in the development process. The design documentation may include writing the application programming interface (API).
5. Deployment and maintenance: The software is deployed after it has been approved for release.
6. Maintaining: Software maintenance is done for future reference. Software improvement and new requirements (change requests) can take longer than the time needed to create the initial development of the software.

**Agile**

## Agile project management is an iterative approach to planning and guiding project processes.

Agile development is a phrase used in software development to describe methodologies for incremental software development. Agile development is an alternative to traditional project management where emphasis is placed on empowering people to collaborate and make team decisions in addition to continuous planning, continuous testing and continuous integration.

Following are the Agile Manifesto principles

* **Individuals and interactions** - in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
* **Working software** - Demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.
* **Customer collaboration** - As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
* **Responding to change** - agile development is focused on quick responses to change and continuous development.

**Waterfall**

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Some situations where the use of Waterfall model is most appropriate are:

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.
* The project is short.

|  |  |
| --- | --- |
| **Agile** | **Waterfall** |
| 1. Agile accepts change and backward scalability is possible | 1. Under waterfall approach we cannot change the decisions and implementations that we had made under the previous stages. If we want to make changes under waterfall we will have to build the entire project from the scratch once again. |
| 1. Very flexible | 1. It is structured |
| 1. Many small projects | 1. One big project |
| 1. Iterative/cyclic | 1. A sequential process |
| 1. Best for those who want continuous improvements | 1. Suited for situations where change is uncommon |
| 1. A process in which requirements are expected to evolve and change | 1. A process that requires clearly defined requirements upfront |

**Iteration:**

An iteration, in the context of an Agile project, is a timebox during which development takes place, the duration of which may vary from project to project. In [agile software development](http://searchsoftwarequality.techtarget.com/definition/agile-software-development), an iteration is a single development cycle, usually measured as one week or two weeks. An iteration may also be defined as the elapsed time between iteration planning sessions.

**Scrum:**

[Scrum](http://www.mountaingoatsoftware.com/agile/scrum) is an agile process most commonly used for product development, especially software development. Scrum is a project management framework that is [applicable to any project](http://blog.mountaingoatsoftware.com/deciding-what-kind-of-projects-are-most-suited-for-agile) with aggressive deadlines, complex requirements and a degree of uniqueness. In Scrum, projects move forward via a series of iterations called sprints. Each sprint is typically two to four weeks long.

SCRUM is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

Scrum is:

• Lightweight

•  Simple to understand

•  Difficult to master

## **Kanban**

Kanban developed as a subcomponent of the Toyota Production System and has its origins in these Lean and Just In Time (JIT) manufacturing processes.

In Kanban the workflow is visualized: work is broken down into small, discrete items and written on a card which is stuck to a board; the board has different columns and as the work progresses through different stages (e.g. ready, in progress, ready for review etc) the card is moved accordingly.

In Kanban the number of items that can be in progress at any one time is strictly limited.

The average time it takes to complete an item (sometimes called the ‘cycle time’) is tracked and optimized so that the process becomes as efficient and predictable as possible. The elimination of waste is paramount.

## **XP**

XP is short for eXtreme Programming, a framework which focuses heavily on ensuring the quality of delivered software and which prescribes engineering solutions towards that end.

An XP team (comprised of all who contribute to the project) engage in Release Planning and Iteration Planning. They work in very short development cycles so that changes requested by the customer (who works on-site with the team) can be incorporated frequently.

Through more than a dozen core practices which include Test Driven Development, Customer Testing, Continuous Integration, Small Releases and Pair Programming, XP works towards a continuously improving, high quality product which can respond to changes in customer requirements.

**V-Model**

The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model.

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

**Spiral**

The spiral model is similar to the incremental model, with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral, starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral. It’s one of the software development models like Waterfall, Agile, V-Model.

**RUP**

The Rational Unified Process attempts to capture many of modern software development's best practices in a form suitable for a wide range of projects and organizations. This process recognizes that the traditional waterfall approach can be inefficient because it idles key team members for extended periods of time. Many feel that the waterfall approach also introduces a lot of risk because it defers testing and integration until the end of the project lifecycle. Problems found at this stage are very expense to fix.

**Business Analysis**

Business Analysis is the practice of enabling change in an organizational context, by defining needs and recommending solutions that deliver value to stakeholders.

Business analysis is the business process of assessing an organization’s structure, processes, technology, and capabilities to identify and define solutions to roadblocks that impede the achievement of organizational goals. It enables adaptation in an ever-changing business and regulatory environment to allow the organization to grow in the manner defined by management.

**Business Analyst**

A business analyst is any person who performs business analysis activities, no matter what their job title or organizational role may be. Business analysis practitioners include not only people with the job title of business analyst, but may also include business systems analysts, systems analysts, requirements engineers, process analysts, product managers, product owners, enterprise analysts, business architects, management consultants.

Many other jobs, such as management, project management, product management, software development, quality assurance and interaction design rely heavily on business analysis skills for success.

**Day to Day activities of Business Analyst**

A business analyst's daily "job duties" can vary greatly, depending on the nature of the current organization and project. However, there are some activities that the business analyst will commonly do in the course of every project. They include: investigating goals and issues, analyzing information, communicating with a wide variety of people, documenting findings, and evaluating solutions. For a given project, the business analyst will usually try to define and oversee a series of carefully structured tasks aimed at achieving the general goals of analysis, synthesis, planning, and evaluation. Of course, these tasks are bound to require a flexible approach matching the circumstances.

Let's look at how business analysts spend their day.

**Investigating:** Business analysts spend a good deal of time asking questions. In order to expand their knowledge, they might conduct interviews, read, and observe the operations of a functional business area they've been assigned to. They'll do research and look for solutions options, both within and outside the organization.

**Analyzing:** Business analysts spend a great deal of time analyzing the information they acquire—studying it for patterns and trends; continually reviewing it to ensure that it is current, thorough, and accurate; and probing deeply for the sources of a problem and potential solutions. Many problems suggest multiple solutions, so the business analyst may have to spend lots of time analyzing and comparing solutions and weighing them against the actual needs.

**Communicating:** Business analysts spend many hours actively communicating. More than simply talking, this means listening to verbal and non-verbal messages, establishing open dialogue, confirming that they've understood what they've heard, and communicating what they learn to those who'll build the actual solution.

**Documenting:** Business analysts spend a fair amount of time documenting what they learn and observe, and the results of their analyses. The business analysts must reflect on the best ways to document specific types of information, whether as text or visual form (charts, graphs, illustrations, etc.).

**Evaluating:** A business analyst must also spend time identifying options for solving specific problems, then help select the best one, and then evaluate the chosen solution through its design and construction to ensure that it continues to meet the business needs and that the team chooses the best process for putting it in place.

Reference Link: <http://requirementssolutions.com/what-do-business-analysts-do-2/>

**Sponsor**

A stakeholder who authorizes or legitimizes the product development effort by contracting for or paying for the project.

**Stakeholder**

A group or person who has interests that may be affected by an initiative or influence over it.

**Customer**

A stakeholder who uses products or services delivered by an organization.

**Subject Matter Expert (SME)**

A stakeholder with specific expertise in an aspect of the problem domain or potential solution alternatives or components.

**End User**

A person or system that directly interacts with the solution. End users can be humans who interface with the system, or systems that send or receive data files to or from the system.

**Project Manager**

The stakeholder assigned by the performing organization to manage the work required to achieve the project objectives.

**Tester**

A stakeholder responsible for assessing the quality of, and identifying defects in, a software application.

**Regulator**

A stakeholder with legal or governance authority over the solution or the process used to develop it.

|  |  |
| --- | --- |
| **Agile** | **Spiral** |
| 1. More risk of sustainability and maintenance. | 1. Better risk management |
| 1. Minimum rules, documentation easily employed. | 1. Large number of intermediate stages requires excessive documentation |
| 1. Little or no planning required. | 1. Planning is required. |
| 1. Easy to manage. | 1. Management is more complex |
| 1. Early delivery of partial working solutions. | 1. End of project may not be known |
| 1. Suitable for small projects. | 1. Not suitable for small or low risk projects. |
| 1. Depends heavily on customer interaction | 1. Does not depend heavily on customer interaction |
| 1. Every iteration is a separate model | 1. Every iteration is not a separate model |
| 1. Agile is the combination of iterative and incremental | 1. Spiral is the combination of iterative and waterfall |

|  |  |
| --- | --- |
| **Waterfall** | **V-Model** |
| 1. Waterfall model is traditional where Testing comes in last phase | 1. V model represent verification and validation |
| 1. Waterfall Model the tester role will take place only in the test phase | 1. V-Model role will take place in the requirement phase itself |
| 1. Waterfall madel is a fixed process u can't make any changes in the requirement or in any phase | 1. V-Model u can make any changes in the requirements |
| 1. Waterfall model used only the requirements are fixed | 1. V-model can be used for the any type of requirement |
| 1. We can say waterfall model is mostly we use in product based application | 1. For project based application V model is mostly used. |