**SYSTEM DEVELOPMENT LIFE CYCLE**

System Development Life Cycle is a project management model that defines the stages involved in bringing a project from inception to completion. SDLC Models includes: Waterfall, Spiral, Iteration and Agile processes. One of the most significant consideration when building application is Testing and security of the application.

The seven cycles are as listed below:

Step 1: Planning -

Step 2: System Analysis -

Step 3: System Design -

Step 4: Development -

Step 5: Implementation -

Step 6: Integration and Testing -

Step 7: Operations and Maintenance –

One of the most commonly used Models of SDLC is agile:

## What is Agile?

The [Agile Manifesto](http://agilemanifesto.org/) formally introduced the idea of Agile Software Development in 2001. Agile is a collection of software development methods used by groups of developers to quickly develop and continuously improve software. The Agile methodology focuses incremental and repeatable development, in which solutions are shaped through the cooperation of organic, cross-functional teams.

The Agile process promotes constant adaptation to feedback and change. By time-boxing segments of work into short periods known as “sprints,” this framework promotes tight iterations for the entirety of the [Software Development Life Cycle (SDLC)](https://www.veracode.com/security/software-development-lifecycle-sdlc). At the end of each sprint, the Agile methodology calls for a shippable product that could at least theoretically go to the customer, although it is often the case that for a full update to go out the development team may have to go through several sprints.

The Agile manifesto spells out 12 principles for agile developers:

1. Highest priority is to satisfy the customer
2. Welcome changing requirements, even if they are late in development
3. Deliver working software frequently
4. Close, daily cooperation between business people and developers
5. Build projects around motivated individuals and trust them to complete it
6. Face-to-face conversation is the best form of communication
7. Working software is the principal measure of progress
8. Sustainable process, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances

## Benefits Agile Software Development

Agile came largely as a response to the flaws recognized in software development process that preceded it. The most common, waterfall, was heavily front loaded and focused on developing a long term development plan followed by the implementation of that plan. Agile, on the other hand, focuses on cutting a project into short segments and promoting self-organizing groups.

Proponents of Agile believe that the process helps to better satisfy customer needs as a result of its incremental development, efficient communication, constant adaption and superior quality of product:

**Incremental development**

Agile methods minimize planning and break tasks into small increments placed in short time frames (time-boxes) that typically last from one to four weeks. Self-organizing, cross-functional teams working in all functions: planning, analyzing requirements, design, coding, and testing. Once completed, the working product is demonstrated to all those involved.

This incremental approach minimizes overall risk and allows for quick adaptation to a changing environment, request or unforeseen issues.

**Efficient, face-to-face communication**

Each agile team includes a customer representative, chosen by stakeholders to act on their behalf and makes a personal commitment to being available for developers to answer mid-iteration questions. At the completion of each development cycle, customer representatives meet with stakeholders to review the progress made and ensure that they are in alignment with the customer’s needs and company’s goals.

Often times an “information radiator” either in the form of a physical display or project tracking software such as Jira is used to display an up-to-date summary of the status of a project.

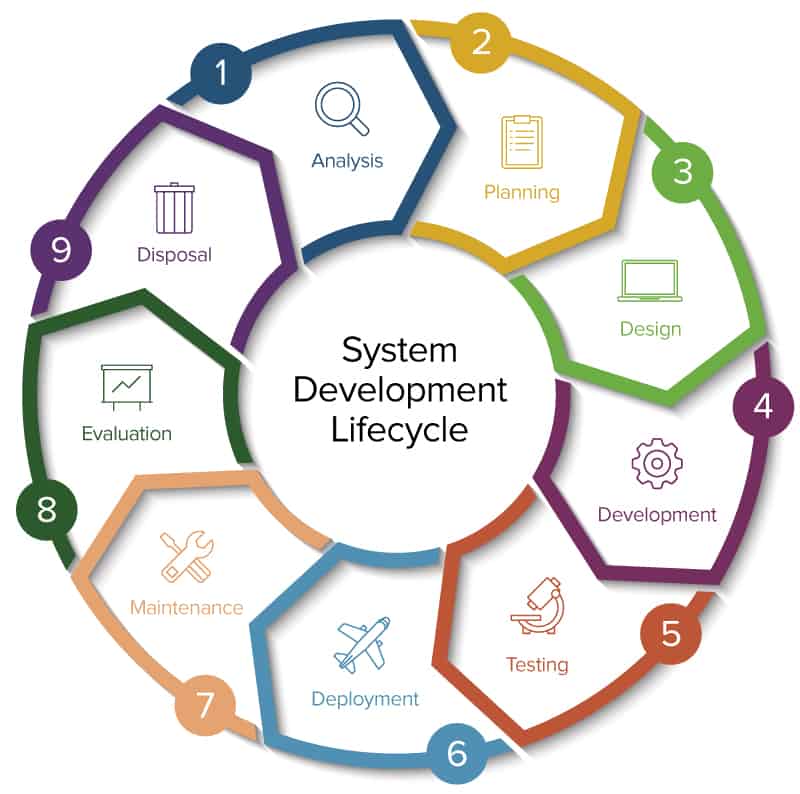
**Short feedback loop and adaptation to change**

Most groups who use Agile have daily status meetings i.e. stand-ups" or “daily scrums.” In this short session, typically 10 to 15 minutes, team members report what their work from the day before, what work they plan on doing today, and what roadblocks are impeding their task completion. This “inspect-and-adapt” approach reduces development costs and time-to-market, simultaneously stakeholders have recurring chances to test and improve releases to be ready for the real world.

**Focus on quality**

Specific tools and techniques, such as continuous integration, pair programming, test-driven development, design patterns, automated unit testing, domain-driven design, code refactoring, [static analysis](https://www.veracode.com/products/binary-static-analysis-sast) and other techniques are often used to improve code quality. The key to assuring this quality during the Scrum process is to rely on automated systems; manual approaches take too long to keep pace with the high paced clip of the Agile development process. Organizations wishing to attest to the security of the applications they develop using the Agile process turn to Veracode to quickly find flaws in the code that they wrote the day before.

**Diagramatical Representation of SDLC:**



Summary of LAMP Technology:

A LAMP stack is **a bundle of four different software technologies that developers use to build websites and web applications**. LAMP is an acronym for the operating system, Linux; the web server, Apache; the database server, MySQL; and the programming language, PHP.

