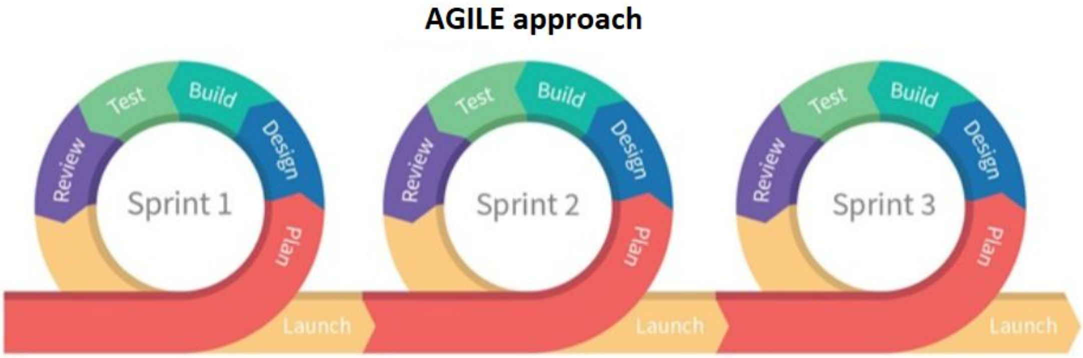
Agile **approach of Software Development**

Software industry has been continuously evolving both in terms of technology used and processes used for building great software to serve millions of customers across the globe. It has become a mandate to have 100% satisfied customers by delivering the best software with precise quality (meeting the business needs), in quick time as possible, and with desirable cost.

The traditional ways of building software could no longer be used, and thus resulting into a needs for a dynamic way of software development, which ensure best possible software delivery managing the balance between customer needs and technical delivery. So as to have a successfully functional over all system. This emerged AGILE.

What is AGILE?



- Agile is a ‘**time-box, iterative approach**’ of software development.
- *Time-box* - within a specified time frame you need to develop and release the software functionality. Ideal time frame would be 1 - 4 weeks, to have a working ‘ready’ piece of software.
- So, you don’t wait for building up entire software and then do a bulk release in one go.

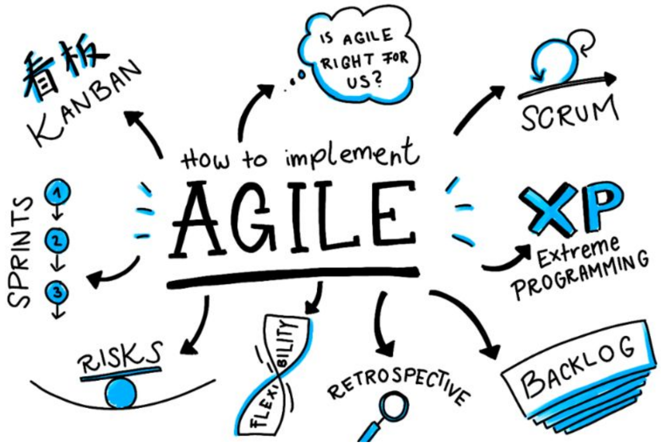
- This would help us to know, whether our understanding of the requirement is same as what the client wants.
- Requirements and discussion that we have had are aligned to what the customer wants.
- In every iteration (called as Sprint – in Scrum modal), you *plan*, *design*, *build*, *test*, *review*, *release* a working piece of software.
- So, more the iterations that get carried out, the customer gets to know what is building up. And can strategize his business accordingly.
- To overcome this gap (gap is, customer is seeing the developed component, in the very end of the software development), and to involve the customer more in the S/W development cycle itself, we introduced AGILE. This results in 100% customer satisfaction.
- Old method, generally clients were involved in the very early stages and were not interacting anymore, just waiting on the final outcome. Due to this, clients had no idea about what is going to be delivered to them in the end. Also, earlier methods used to freeze client's requirements after a particular time and so customers were not able to change or add any features flexibly. Such restrictions started affecting the IT industry and leaving customers frustration and not getting what they want, despite putting enough money.

Advantages of AGILE method of software development

- Good for the clients:
    - Client changes can be incorporated during the SDLC itself.
    - Client get a chance to see what is getting developed during the development process itself no need to wait till the very end for getting the product.
    - Helps ensuring the organization to get the confidence of what we are developing is equivalent to what is expected by the client.
    - We can develop and deliver any new requirement to the client, even though the development has already started.
    - It helps to ensure 100% customer satisfaction.
    - Clients can market their software and get started with the business as soon as first key delivery is done. And not wait for full completion of software.
- Good for employees:
    - The process ensures employees to estimate and own their work.
    - It helps them being responsible for their work.
    - Deliver with high productivity.
    - Be a great team player, and even help others getting their work done.

    - Work as a unit, instead of a independent developer.
    - Generate decision making and result oriented outcome from employees.
    - Chance for choosing to work on different challenges and technologies. Learning opportunities.
- Good for the organization:
    - Having satisfied, re-occurring clients.
    - Having great working employee base.
    - Is a massive benefit for the IT companies.

How to implement AGILE:



There are multiple ways of a applying agile. And market if full of options of which one approach to follow to precisely apply Agile to software development.

However, as far my experience goes, the right way of applying Agile is a mix of ideas and concepts prescribed in such multiple methods. However there are some methodologies who have grown very vastly adapted widely across the industry. For example ‘Scrum’, ‘Kanban’, in proper composition have achieved remarkable results. However further mixing with other methods like ‘Extreme programming’ and ‘Lean programming’, would enhance the strategy and outcome further.

In my view the right way of applying agile, is to have a open mind, so as to understand the problems encountered in software development, and improve on those problems to maximize the delivery. It might even include using the right mix of traditional waterfall approach and current Agile approaches. But reaching to this decision making evolves with time.

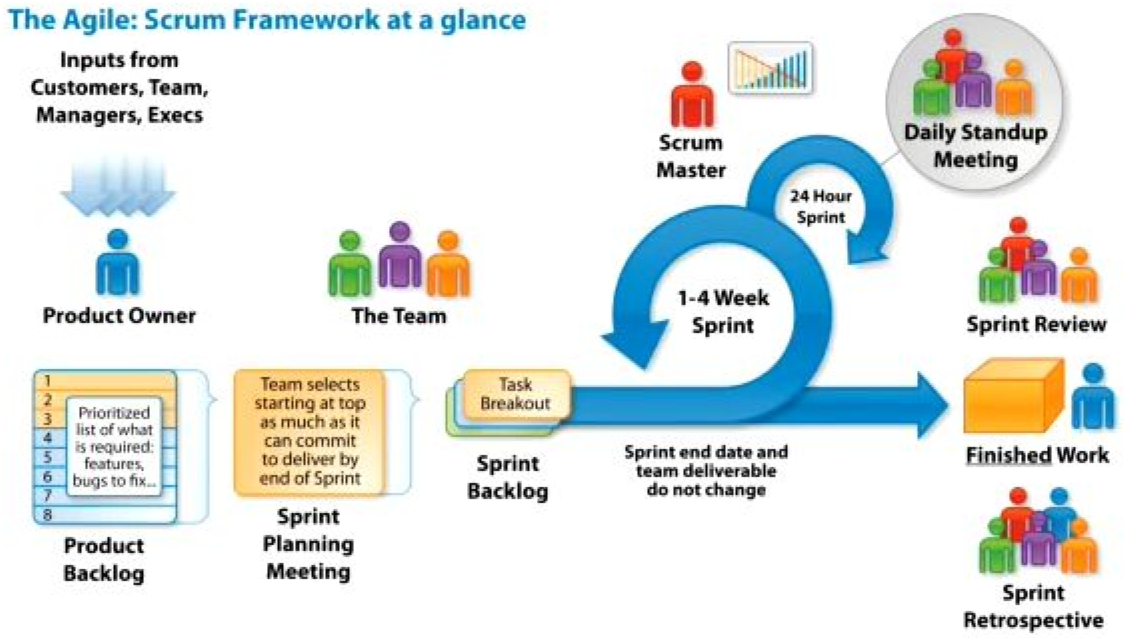
Some popular Agile methodologies include:

- Agile Scrum Methodology
- Kanban
- Extreme Programming (XP)
- Lean Software Development
- Crystal
- Dynamic Systems Development Method (DSDM)

- Feature Driven Development (FDD)

Agile Scrum Methodology

Scrum is a lightweight Agile project management framework that can be used to manage iterative and incremental projects of all types. It has become increasing popular over the years due to its simplicity, proven productivity, and ability to incorporate with various other methodologies. “Scrum” is one of the most widely used framework these days.


The Agile: Scrum Framework at a glance
Inputs from Customers, Team, Managers, Execs
Product Owner
The Team
Scrum Master
Daily Standup Meeting
24 Hour Sprint
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Prioritized list of what is required: features, bugs to fix...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
Sprint Retrospective

SCRUM Methodology:

- Scrum approach divides the s/w development into small chunks called as sprints.
- Each sprint delivers a working piece of software.
- It has multiple designations like, ‘Product owner’, ‘Scrum master’, ‘Team’. The role of traditional ‘Project manager’, is split between ‘Product owner’ and ‘Scrum master’.
- There are tools like Jira (by Atlassian), which provides end-to-end capability to managing projects using Scrum.

Approach

- **“Sprint”**: Entire timeframe is broken down into smaller delivery windows calling as ‘Sprint’. Each sprint delivers working piece of software.
- **Product Backlog**:
  - Product Owner works closely with their team, customer, executives to identify and prioritize system functionality by creating a Product Backlog. The *Product Backlog* consists of whatever needs to be done to successfully deliver a working software system, including features, bug fixes, non-functional requirements, etc.
  - Generally these requirements are specified in the form of user-stories and tracking using some software like Jira.
  - The items are prioritized with most important items on the top and less important at the bottom, which is subjected to change after each sprint or iteration. Goal is to bring focus on what brings values to client's business.
- **Sprint planning meeting**:
  - This backlog is studied to identify; how much work can be delivered at the end of that sprint. Consider the items based on their prioritize and commit to delivery.
- **Sprint backlog**:
  - The team comes together to breakdown requirements into smaller task which can be ready for development. The developers can make a choice to select the work of their choice by discussing with entire team and meeting the promised sprint delivery.
- **Executing the Sprint & Daily standups**
  - Every day team would come together for daily stand up meeting called as scrum, to discuss what was done yesterday, what’s the plan for today, and whether there are any dependencies or blockers in work.
  - This activity is highly important to ensure, that the team meets the planned timelines for sprint delivery.
- **Scrum Master**:
  - Considering the criticality of meeting sprint delivery, ‘Scrum master’ is a designated role, which monitors and works closely with entire team to help solving any troubles which risks sprint delivery.
  - Does appropriate decision-making.
  - Support the team in all possible means to accomplish the task.
- On final scrum day, we have 2 additional meetings
  - **Sprint review/demo**
    - Intended to show to the client and the product owner what the team accomplished.

-
  -
    - Feedbacks from client and project owner welcomed.
    - Ensuring the whatever the client wants, the same software is build and being delivered.
  - **Sprint retrospective**: Would track all the possible blocking/troubles which team faced during in that sprint. This helps to address all such problems and smoothen the delivery process.
- At this time, if the product is completed it would be released
- **Finished work**: piece of product which is completed and functional, after every end of sprint we get it.
- Next goes back to the sprint planning

Note there are dedicated certifications available, for becoming, Scrum team developer, scrum team – Scrum master, scrum team – product owner. But the better you get these, the better overall IT professional you would become.

For more details:

- https://www.atlassian.com/agile
- https://www.scrumalliance.org/about-scrum/overview