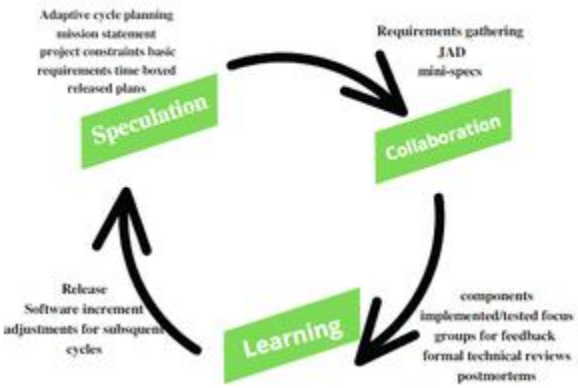
# Adaptive Software Development (ASD)

**Adaptive Software Development** is a method to build complex software and system. ASD focuses on human collaboration and self-organization. ASD **"life cycle"** incorporates three phases namely:

**1.** Speculation
**2.** Collaboration
**3.** Learning

These are explained as following below.




***Adaptive Software Development***

**1. Speculation:**

During this phase project is initiated and planning is conducted. The project plan uses project initiation information like project requirements, user needs, customer mission statement, etc, to define set of release cycles that the project wants.

**2. Collaboration:**

It is the difficult part of ASD as it needs the workers to be motivated. It collaborates communication and teamwork but emphasizes individualism as individual creativity plays a major role in creative thinking. People working together must trust each others to

- Criticize without animosity,
- Assist without resentment,
- Work as hard as possible,
- Possession of skill set,
- Communicate problems to find effective solution.

**3. Learning:**

The workers may have a overestimate of their own understanding of the technology which may not lead to the desired result. Learning helps the workers to increase their level of understanding over the project.

Learning process is of 3 ways:

1. Focus groups
2. Technical reviews
3. Project postmortem

ASD's overall emphasis on the dynamics of self-organizing teams, interpersonal collaboration, and individual and team learning yield software project teams that have a much higher likelihood of success.

# FDD:

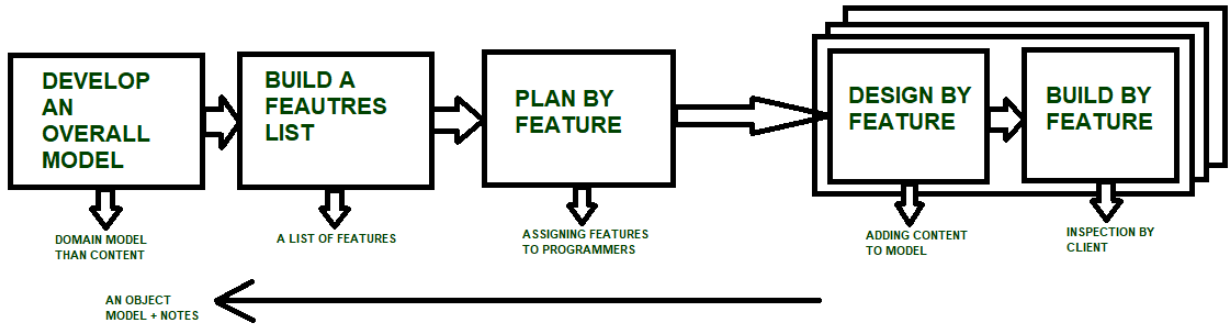
**FDD** stands for **Feature-Driven Development**. It is an agile iterative and incremental model that focuses on progressing the features of the developing software. The main motive of feature-driven development is to provide timely updated and working software to the client. In FDD, reporting and progress tracking is necessary at all levels.

***History***

FDD was first applied in the year 1997 on a real-world application by *Jeff De Luca* for large software development with specific needs of 15-month and 50 persons and published as a discussion in book *Java Modeling in Color with UML* in the year 1999.

***FDD Lifecycle***

- Build overall model
- Build feature list
- Plan by feature
- Design by feature
- Build by feature




***Characteristics of FDD***

- **Short iterative:** FDD lifecycle works in simple and short iterations to efficiently finish the work on time and gives good pace for large projects.
- **Customer focused:** This agile practice is totally based on inspection of each feature by client and then pushed to main build code.
- **Structured and feature focused:** Initial activities in lifecycle builds the domain model and features list in the beginning of timeline and more than 70% of efforts are given to last 2 activities.
- **Frequent releases:** Feature-driven development provides continuous releases of features in the software and retaining continuous success of the project.

***Advantages of FDD***

- Reporting at all levels leads to easier progress tracking.
- FDD provides continuous success for larger size of teams and projects.
- Reduction in risks is observed as whole model and design is build in smaller segments.
- FDD provides greater accuracy in cost estimation of the project due to feature segmentation.

***Disadvantages of FDD***

- This agile practice is not good for smaller projects.
- There is high dependency on lead programmers, designers and mentors.
- There is lack of documentation which can create an issue afterwards.