













- Its Functionality can be developed rapidly and demonstrated.
- Its Resource requirements are minimum.
- It is Suitable for fixed or changing requirements
- It delivers early partial working solutions.
- It is Good model for environments that change steadily.
- Its Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required
- Easy to manage
- Gives flexibility to developers
- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face-to-face conversation is the best form of communication.
- Close, daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed

#### *Disadvantages*

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.
- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is lack of emphasis on necessary designing and documentation.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.

- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

## 5. Future Work

A possible area for future research is the analysis of the effects of process models on mobile-centric startups. Practices such as continuous integration in Agile or continuous deployment in Lean Startup become nearly impossible in a startup with heavy focus on mobile development. The process models that focus heavily on the ability to integrate and deploy continuously or split testing may not be effective for mobile-centric startups. In this new era in which mobile development is becoming more and more popular, perhaps a new process model is required.

## 6. Conclusion

The Agile Manifesto is considered as a milestone for the development of new agile methodologies, since it provided a general base with its four values pointing out the most crucial facts of a software development process.

The process or role oriented traditional methods have difficulties, since individuals are not, in fact, replaceable; especially where the individuality is important as in software development. Furthermore, rather than focusing on strict and long-term plans, Agile Software Development focuses on responding to changes and on the working software with less documentation. Even though some agile practices are not new, agile methods themselves are recent and have become very popular in industry. There is an important need for developers to know more about the quality of the software produced. Developers also need to know how to revise or tailor their agile methods in order to attain the level of quality they require. The conclusion we draw here is:

1. Agile methods do have practices that have QA abilities, some of them are inside the development phase and some others can be separated out as supporting practices
2. The frequency with which these agile QA practices occur is higher than in a waterfall development
3. Agile QA practices are available in very early process stages due to the agile process characteristics.

**Table 2:** Provides a comparison among different discussed methodologies and quality assurance factors obtained from them in different processes. Pragmatic Programming has been intentionally left blank due to its immature nature and unavailability in literature.

