

## Build your Project using Agile Methodology

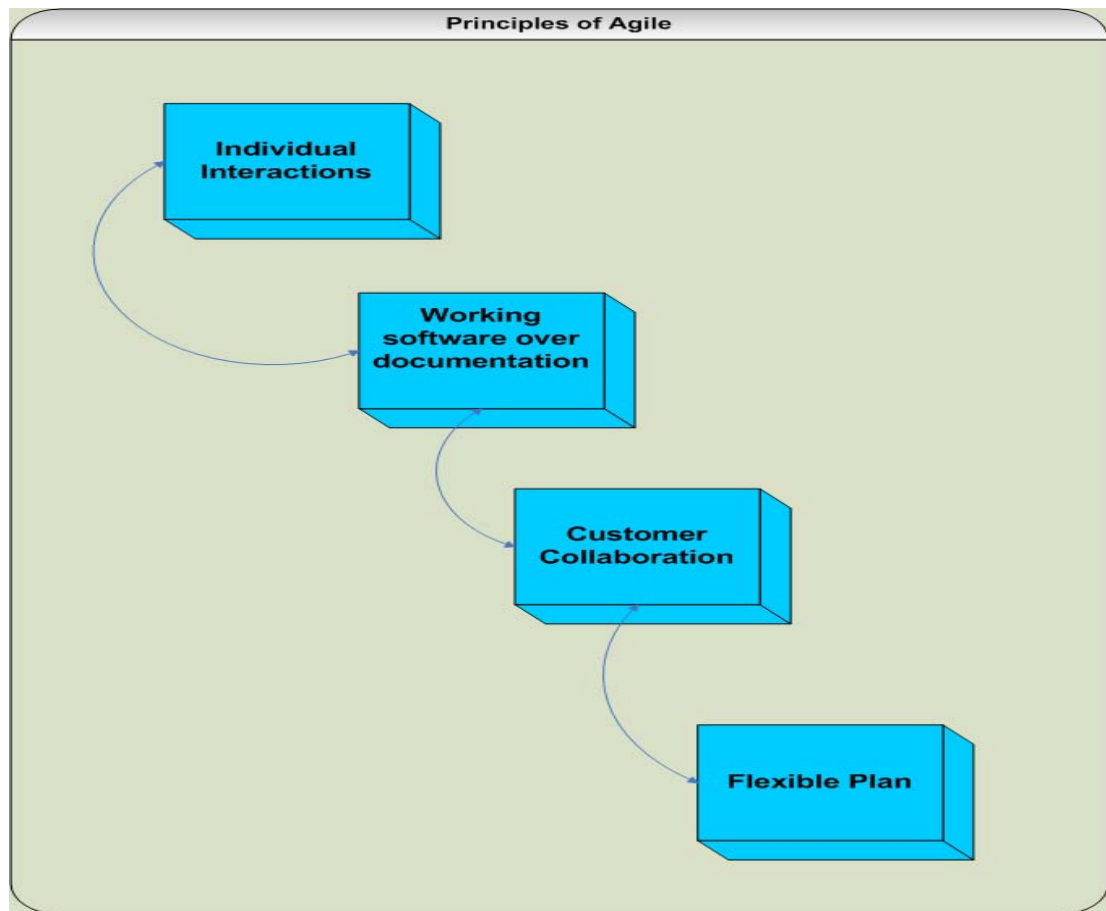
#1 of a Series, by Pavan Kumar Gorakavi, M.S., M.B.A, G.M.C.P, C.A.P.M.

### 1. What is Agile Methodology?

Agile denotes nimbleness, a light-weight systems development methodology, based on iterative development where solutions evolve from tightly collaborated cross-functional teams. In early 2000, many people were having difficulty identifying a methodology that delivered a product quickly and with good responsiveness. A group of Industry experts formed an alliance called an 'Agile Alliance'. Over two days they worked to create statement of value, which results in the manifesto of agile alliance. Over next three months they formulated the principle of agility.

Agile movement observes the light of a new day with an *Agile Alliance Manifesto*. Key principles of agile are illustrated in figure 1. This manifesto primarily focuses on the following aspects:

- 1) Individuals and interactions over process and tools
- 2) Working software over comprehensive documentation
- 3) Customer collaboration over contract negotiation
- 4) Respond to change over following a plan.[1]

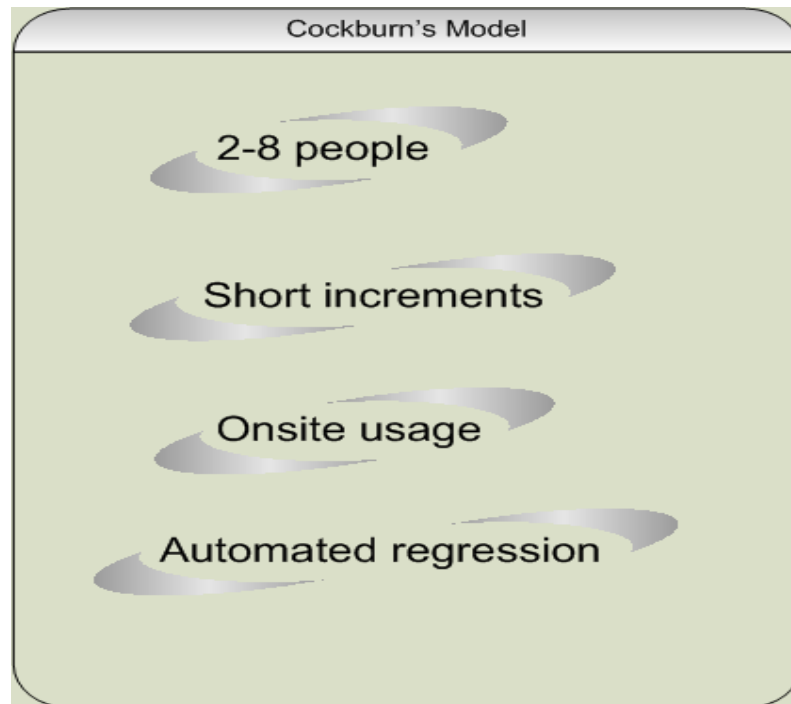


**Figure 1: Key principles of Agile**

Human resource management plays a vital role in making a project successful. Good human resource management and a high degree of interaction between the resources can improve process success. Even a group of strong players can fail in yielding the results, if the degree of communications or team play fails. Excessive documentation or limited documentation can hamper the success of the project. Optimal documentation has to be provided as a landmark for project guidelines. Successful projects can be implemented when customer is collaborating during contract negotiation. Successful projects involve customer feedback on a regular and frequent basis.

In Agile methodology, customers works closely with the development team, providing frequent feedback, which results in developing as-expected product. Agile methodology provides an ability to respond to change that determines the result of the project. A good strategy with good planning helps in building a robust system. An agile system should be robust and flexible enough to accommodate change in requirements at later part of the game.

Cockburn (2) states agile methodology as light and sufficient rules of project behavior which practices human and communication oriented rules. Cockburn proposes 'sweet spots' which enhance the outcome of the project.



**Figure 2: Cockburn Model**

Miller (3) outlines the agile software process with the following set of characteristics.

- Modularity on development process level.
- Iterative with short cycles enabling fast verification and correction.
- Time bound with iteration cycles from one to six weeks.
- Parsimony in development process removes all unnecessary activities.
- Adaptive with possible emergent new risks.
- Incremental process approach that allows functioning application building in small steps.
- Convergent and incremental approach minimizes the risks.
- People oriented.
- Collaborative and communicative working style.

Highsmith and Cockburn report that the software development process is directly impacted by a changing environment. Satisfying customers at the time of delivery has taken precedence over satisfying the customers at the moment of project initiation. According to Cockburn, agile methods are designed to:

- Produce the first delivery in weeks, to achieve an early win and rapid feedback.
- Invent simple solutions, so there is less to change and making those changes is easier.
- Improve the design quality continually, making the next iteration less costly to implement, and
- Test constantly, for earlier, less expensive, defect detection.

Ambler [4] summarizes agile methodology as a people matter, with less documentation, communication providing a pivotal role, modeling tools are not very useful, and big upfront design is not required.

Boehm[5] analyzes agile methodology as component module completion with agile knowledgeable developers, dedicated and collaborative customers, rapid change requirements, inexpensive re-factoring and smaller teams and product, with a primary objective of 'Rapid Value'. He made some significant analysis of agile methods, open source software, and plan-driven methods.

In a nutshell, we can summarize agile methodology as an incremental process, which yields successful product in a highly communicated dynamic environment.

## 2. Principles of Agile Methodology

Agile methodology has a primary focus on Individuals and their communication, Working Software over comprehensive documentation, Customer collaboration over contract negotiation, and Responding to Change over following a plan. We can summarize following principles that differentiate an agile process.

### a) *Customer Satisfaction*

An agile process helps the customer to have a quick view of the product. Agile process delivers early and often. As a rudimentary system is provided within the first couple of weeks, Customers have a chance to make requirement changes if they feel that what they see doesn't match what they envision. Requirements change dynamically but they strictly adhere to deadlines.

### b) *Good team with higher accountability*

As agile methodologies are rapid in nature, a team of good players needs to be accommodated. Though all players need not be strong players, players with higher accountability and user involvement is imperative. The team should consist of developers, customers, and any other business related personalities. The team should be empowered to make decisions.

### c) *Light weight requirements, flexibility for rapid changes*

Requirements have to light weight. We can expect change in requirements late in the development process. It should be noted that requirements evolve with a fixed time scale. This differentiates agile methodology with conventional development methodology. Generally industries go with 80-20 acceptance policy.

### d) *Small and Iterative cycles*

An "Eating an elephant, one Bite at a time" policy should be implemented when we focus on agile methodology. Agile methodology advises to develop each project in a small size with an incremental nature. Agile is iterative in nature. This nature reduces the risk of *what you see vs. what you envision* aspects. It also provides effective cost management, and less re-factor cost. Deliverable are given preference over documentation.

e) *High degree of communication*

There should be a high degree of communication between the team members in order to implement agile methodology effectively. There should be significant interaction between the customers, developers and stakeholder. Just-In-Time [JIT] approaches must be implemented for successful practice of agile methodology.

f) *Frequent releases/deliveries*

An agile methodology is incremental in nature. A rough draft will be available to customers within the first couple of weeks. The agile cycle is at most 2-3 months in size. Sometimes they are even less than that. Agile encourages an incremental delivery system with shorter cycles of development. Cycle size varies based on consensus among stakeholders, size of the project, and environment of the project.

g) *Early Test, Early feedback*

An agile team includes developers, testers, and stakeholders. As testers are an integral part of the development team, unlike traditional methodology, testing can be performed in synch with development. Early product delivery yields early feedback, which opens an option to compare what customers see vs. what they envision. This kind of approach reveals implementation difficulties at an initial stage, which provides an opportunity to renegotiate requirements.

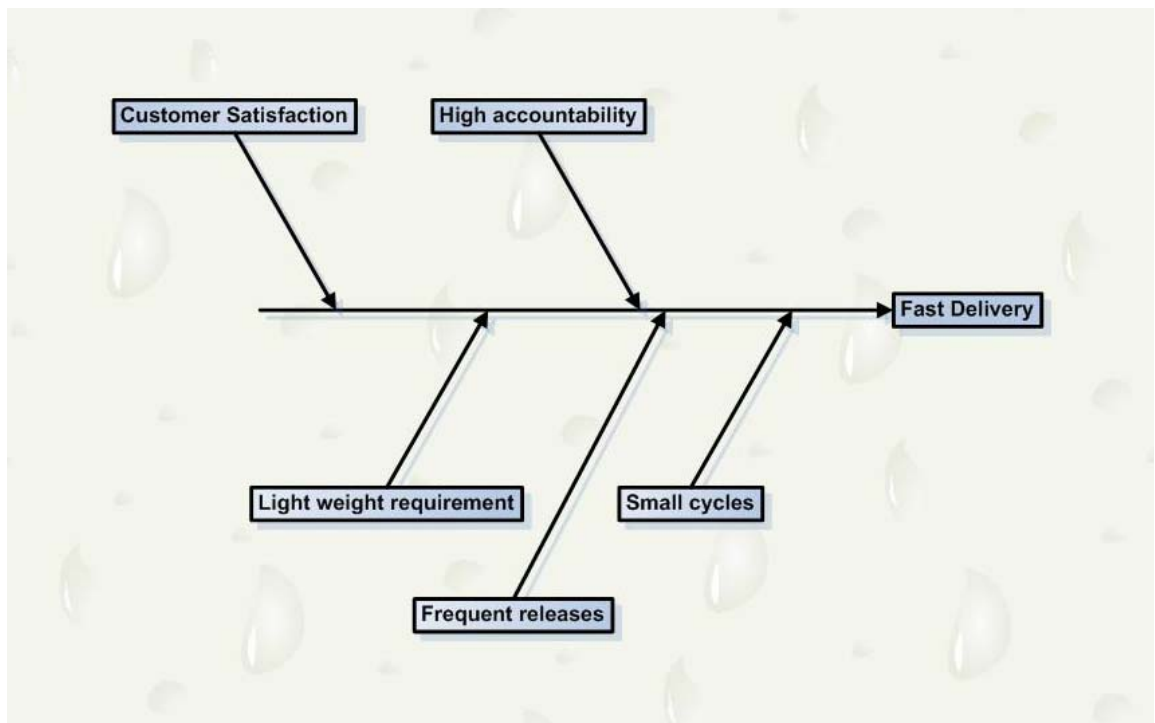


Figure 3: Principles of Agile

### 3. Benefits of Agile Methodology

Agile methodology provides many benefits for end users. As agile methodology provides rapid delivery in a dynamic environment, this incremental nature provides *cost-effective behavior* for product development. This methodology not only enables to deliver a cost effective prototype, but also provides flexible re-factoring. In a competitive environment, business always competes with others business by releasing innovative products in a dynamic , rapid-changing world, which can be completely supported in agile methodology.

As agile methodology delivers products in a small life cycle the risk of handling a project is optimized. As the product structure is foreseen at a very early stage of the product, the level of structural variation most of the time will be below the tolerance level. As testing team and stakeholders are also part of day-to-day activities, quality of the project is relative higher and sticks to core requirements. As the development goes in a fixed duration with fixed timescales, this helps stakeholders to have cost control on the complete process of product development. Due to the high degree of collaboration and interactions, agile methodology promises the right product delivery, good customer satisfaction, and a challenging environment for the agile team.

#### 4. Agile Practices

There are lots of practices that are primarily based on agile methodology. This section deals with some of the industry's stellar examples: Extreme programming, SCRUM methodology, Crystal family, Feature driven development, and Open source software development are some of its practices.

##### a) Extreme Programming

Extreme programming has evolved as a substitute for traditional development methodologies. The life cycle of extreme programming often has 5 phases. They are: *Exploration, Planning, Development, Maintenance and Death*. Use case stories are written in the exploration phase, priorities of the use cases are identified in planning phase, products are delivered in small cycles during iteration to release, the productionizing or development phase includes extra testing, new changes and their priorities, and the final phase of the methodology is death, when the customers no longer have any stories. Extreme programming teams include programmers, customers, testers, trackers, coach, consultant and managers.

Extreme programming encourages small/short releases, simple design, pair programming, collective effort and high accountability. Extreme programming actions and use of the methods must be customized based on the project need.

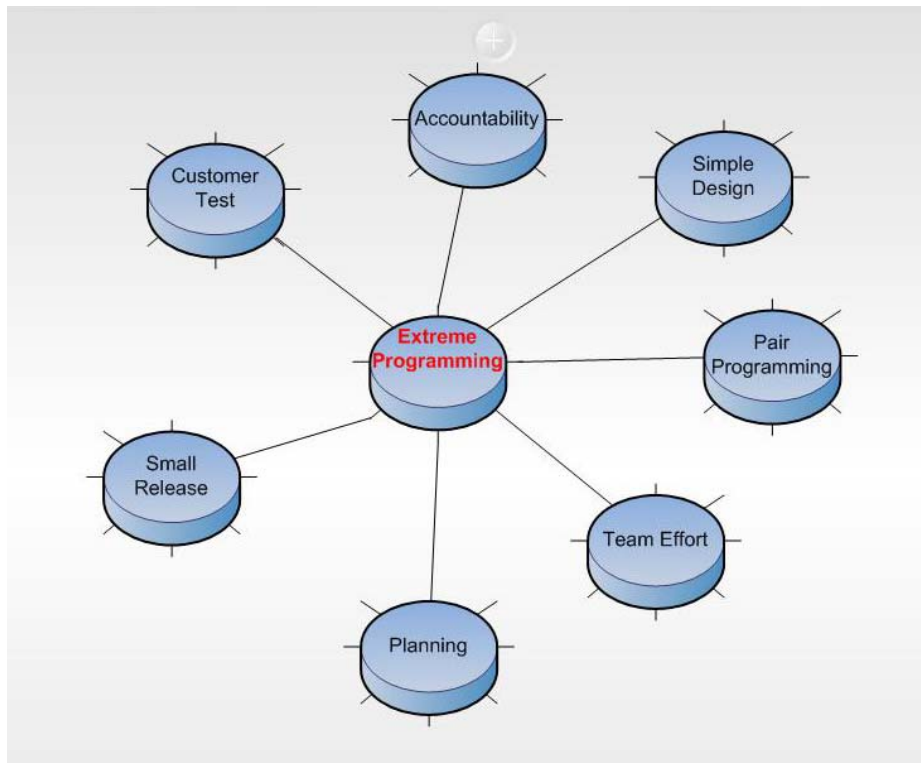


Figure 4: Active players in Extreme Programming

**b) Scrum Methodology**

Initial references to ‘Scrum’ point to the article of Takeuchi and Nonaka [6], in which they demonstrated a quick, adaptive, self-organizing product development process in Japan. This process was developed to address system development. The Scrum methodology includes three phases: Pre-game, game, and post-game. Project day-to-day activities are tracked using burn down charts.

Scrum implementers include Scrum master, developers, testers, product owners, customers and management. Sprint Backlogs have all required features to be fulfilled. After prioritizing the tasks in a sprint planning meeting, tasks are developed in a given sprint. Usually the sprint size varies from 4-12 weeks. This process is illustrated in figure 5.



**Figure 5: Scrum methodology**

More articles in this series will continue to explore the Agile Methodologies, their primary characteristics, and their strengths.

**Conclusion**

Agile methodologies have numerous trend-setting features that make them stand out over conventional planning systems. Agile software can be implemented if the scope of the project is incremental, cooperative, straightforward and adaptive. And, there are lots of CASE tools available to support your agile processes.

### About the Author

Pavan Kumar Gorakavi is working as a Senior Software Developer in Dallas, TX. He is settled in Dallas, TX with his family (wife Swapna Gorakavi and son Anish Gorakavi). He is VP - programs for *asapm* Young Crew. Pavan earned his Bachelor's degree in computer science from Jawaharlal Nehru Technological University and Masters in computer science from Lamar University. He did his MBA from University of Texas at Dallas and GMCP from Southern Methodist School. Pavan holds SUN, IBM and PM Institute certifications.

Pavan Gorakavi authored a book on 'Artificial Intelligence' published by Rahul publications - India, and 'Digital Electronics' published by Subhash publications, India. His research interests are Artificial Intelligence, Agile methodologies, and Software development in ADA, Prolog and Java. You can reach Pavan at [gorakavi@gmail.com](mailto:gorakavi@gmail.com).



### About this Series

This article is the first in a series by Mr. Gorakavi on Agile, posted on the *asapm* website; watch for the others in the series. And, although the concepts of Agile are most-common in Software Development projects, increasingly Agile and Lean PM methods are also turning up in many other project areas, including Engineering and Manufacturing, where some assert they actually originated.

### References:

- [1] Agile manifesto, [www.agilemanifesto.org](http://www.agilemanifesto.org), <http://www.agilealliance.org/>
- [2] Cockburn. Agile software development, Addison-Wesley
- [3] Miller. G.G. The characteristics of Agile software process, 39th conference of object oriented languages and systems.
- [4] Ambler, S. Agile Modeling: Effective Practices for Extreme Programming and Unified Process, John Wiley & Sons, Inc.
- [5] Boehm, B. Get Ready For The Agile Methods, With Care.
- [6] Takeuchi, H. and Nonaka, I. The New Product Development Game. Harvard Business Review.
- [7] Schwaber, Ken. Agile Project Management with Scrum, Microsoft Press, Redmond, WA, 2004.
- [8] Fleming, C.D., Koppelman, Joel M. Koppelman, Earned Value Project Management – 2nd Edition, Project Management Institute, Newtown Square, Pennsylvania, 2000.
- [9] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, Juhani Warsta. Agile software development methods, VTT Publications.