Chapter III

TRANSITIONING OF TRADITIONAL SOFTWARE DEVELOPMENT METHOD TO AGILE METHODOLOGY

3.1 INTRODUCTION

According to Agile Manifesto [6] ‘individuals and interactions over procedures and tools” is the primary proverb in Agile environment. It means processes; procedures and tools have less value in comparison with individuals and interactions. In traditional approaches like waterfall, spiral, V-model etc. process remains fixed. All the phases are properly described and documented so that anyone can follow this fixed and static approach. The static approach of traditional software development method and dynamic approach of ASD makes this matter debatable as transitioning is taking place from waterfall or any other traditional model to Agile model in most of the software companies.

ASD [71] introduces changes in work habits. When an organization wishes to transition to ASD, a change is required at the organizational level. To ensure this change in a software organization many concerns need to be discussed like upper level management attention, team interest, infrastructure needed and many more [99]. Various approaches have been suggested by different researchers for these organizational changes in general and for transition to Agile software development.

Manns and Rising [54] suggest 48 patterns for change introduction. The patterns are the result of years of documenting observations, investigations from community who have introduced latest ideas, reading various topics of change and finding out how these problems are tackled in history.
Orit Hazzan and Yael Dubinsky [71] proposed an organizational survey for transitioning from traditional to Agile. This survey helps to understand the current situation and status of software development in the organization. It also helps in the decision whether the Agile approach fits for the organization or not. The author also proposed four major categories of cooperation tools in change processes: power, administration, leadership and customs. To choose the right cooperation tool requires assessing the software industry along two significant extents: the degree to which team agree on what they actually want.

D. Leffingwell [15] discussed how Agile methods can be useful to enterprise development. He also discussed seven most excellent practices of agility that are best at the enterprise level. Further his book provides an additional set of seven enterprise capabilities that organizations can master to get the benefits of agility on an enterprise scale.

A critical look at the above literature indicates that the previous approaches of transition do not provide an Agile model with the help of which transitioning can take place. Since most of the organizations are working on a traditional SDLC models, there is a need of mapping of the traditional model into Agile life cycle model so that transitioning can take place between two SDLC models in appropriate manner. An Agile model and a mapping function has been proposed in this research work so that transitioning can be attained with ease of team members and upper management.

3.2 PROPOSED AGILE MODEL

To accept change is the compulsory requirement for the Agile developers. Agile cannot exist in industry without accepting change. But the problem is how to perform the transitioning from traditional model to Agile model when the traditional model is the base of the organization and every team member has expertise in that. For accepting change, in the start, things seem to be very difficult but with the support of top management, scrum master and coach, Agile can be implemented with great success.
This section describes a mapping model for transitioning by considering the existing traditional model of the organization [79,90]. The following are the main components of the proposed Agile model which are shown in Figure 3.1.

- Team Formation by good recruitment policy and good team interaction (TFR)
- Goal Building cycle with business Analyst, quality assurance analyst and customer (GBC)
- Coding and Testing activities with Communication and co-ordination (CTC)
- Budget and Effort estimation (BEE)
- Satisfaction for all parties (SFP)

Figure 3.1: Proposed Agile Model
• Demonstrations in Review with feedback (DRF)

• Risk evaluation and correction (REC)

These seven components are the base of an Agile model. The description of each component is given below:

3.2.1 Team Formation by Good Recruitment Policy, Good Team Interaction (TFR)

In Agile working environment, good recruitment policies should be followed to find the right person. In the Agile team, there can be experienced team members as well as freshers. The attitude of a team member towards work should be the biggest factor while doing recruitment.

![Figure 3.2: Team Formation by Good Recruitment Policy](image-url)

Figure 3.2: Team Formation by Good Recruitment Policy
A experienced Agile team can be formed by upgrading the technical and managerial skills of team by devoting training by trainers, polishing the attitude of team towards work and motivating team time to time.

Figure 3.3: An Agile Team Interaction

3.2.2 Goal Building Cycle with Quality Assurance Analyst, Business Analyst and Customer (GBC)

The user-stories [28] on the basis of requirements are identified and approved by client, quality assurance analyst and business analyst by taking into account the return on investment and market demand. An assessment is approved by finding the competitive stage of the existing products. The presence of quality assurance analyst along with customer helps in setting the pattern in mind so that at the time of pair programming he or she can provide the correct feedback to the developer. Also test cases can be designed before development starts.

3.2.3 Budget and Effort Estimation (BEE)

The budget and effort of a user-story is estimated by considering the various requirements for each user-story after prioritization of user-stories. After initial prioritization and estimation of user-stories, two to three weeks cycle of sprint starts. The
effort estimation in ASD can be done by any famous estimation technique like estimation by analogy or planning poker [60,61,62]. Estimation is possible at three levels namely iteration level, release level and project level. The unit of estimation of user-story is story-points [67,68] and ideal time.

3.2.4 Coding and Testing Activities with Communication and Co-ordination (CTC)

The implementation of story starts when estimation is properly done. In pair programming approach two programmers sit together and work together. One person is the leader who performs coding and testing and second person is the reviewer [87] as shown in Figure 3.4. This approach provides immediate feedback with the help of which number of bugs can be reduced. Otherwise the bugs keep on propagating from one phase to another phase. In distributed pair programming or virtual pair programming or remote pair programming the two programmers work together but they are in different locations.

By pair programming as two developers sit together for coding, so knowledge and programming skills are shared. By this approach of pair programming mistakes are also reduced. Test driven development [14,47] (TDD) approach is also used in ASD in which before writing the code for the user-story test cases are written.
3.2.5 Demonstrations in Review with Feedback (DRF)

At the time of review, Agile team members, upper management, business analyst and customers sit together for demonstrating the software product. Scrum master gives the demo for the product. After demo, the goal matching action is carried out to check that whether story approved is the end product or not. Figure 3.5 shows the feedback system, the feedback can be given by any stakeholder including customer, upper management or any business analyst or existing member of the team. After feedback, a review meeting is done which is an informal meeting between all stakeholders.

![Feedback System Diagram](image)

Figure 3.5: Feedback System

3.2.6 Risk Evaluation and Correction (REC)

Further, in the Agile model, risk assessment is performed for the future user-stories so that risk can be reduced or completely eliminated. In fact customer is not only the customer rather he or she is worried about quality, time and also sustainability of the software product in market for long time. In brief, customer is more concerned about return on investment and benefits [14]. In the proposed Agile model high risk user-stories are detected early so that risk is minimized. If high risk stories are not involved then after-effect of it can degrade the quality of the product.
3.2.7 Satisfaction of All Parties (SFP)

All stakeholders whether Agile team, customers or the upper management are satisfied because final product is delivered on time by Agile processes like continuous working software delivery, continuous feedback from customers, continuous integration and testing and continuous return on investment.

When there is a need for transitioning from existing traditional model to Agile model, the issues that may come during mapping are as below:

- Why transition is needed? Is management or customer interested?
- How transitioning from traditional software development life cycle model to ASD model is performed?
- What will be the mapping function to perform transitioning?
- Whether team is of that much caliber or not? Whether new team is required?
- How effort, time and cost estimation will be done?

After resolving all these above issues, management and team both start work for the mapping function from traditional model to Agile. If some software industry is ready for accepting change then in the start, processes would seem to be very complex but after some time with the support of the team members, organization, upper management and scrum master projects can be implemented using ASD with good success rate. Figure 3.6 shows a mapping function which will be applied when the top management takes the decision for transformation in the organization.
In the expression 3.1 co-ordination effectiveness is proposed which depends upon implicit and explicit factors as shown in Figure 3.7. The expression 3.2, shows the role of mapping function (MF) which is to map the large teams of traditional projects into small and efficient Agile teams (T), long release cycles of traditional models into small sprint (I), large tasks into small stories (J), for pair programming two monitors into one terminal (MO), long feedback cycle into instant feedback (F), estimation in lines of code or functional points into story points (E), extended meetings into daily small meetings (M), late delivery into fast small delivery (D), late testing into test driven testing (TG), and last project manager into no chief or boss approach (B) and self-organized team and co-ordination effectiveness (CE). The implicit factors for co-ordination effectiveness are know why, know what is going on, know what to do and when, know who is doing what. The explicit factors are about right place and right time for doing a particular task.

\[ CE = \text{Implicit factors} + \text{Explicit factors} \]

\[ MF = (T, I, J, MO, F, E, M, D, TG, B, CE) \]
Any of the traditional software development models can be transformed into the Agile model by using this mapping function. In proposed mapping function, ten parameters are there which are must for transitioning to Agile environment in an organization. For team interaction and co-ordination cubicles can be converted into open work surroundings, heavy documentation can be converted into simple story-boards, overtime is converted into 38-40 hrs per week of valuable and effective work. Various automated tools are converted into definite tool for a specific domain. In short, Agile approach is more advantageous with less cost and time.

### 3.3 STEPS FOR APPLYING MAPPING FUNCTION

- Don’t apply the mapping function all of a sudden. Discover the ways to simplify operational and administrative documentation.

- Start with a low risk, small project and develop user stories and scenarios as the feature units and start initial estimations.
• Break the large release cycle to small iterations called as deployment cycles. Change large projects of more than eight months into several versions released after every two months.

• Under the guidance of Agile coach, form the small Agile teams, with experienced people from the different functional areas. Good recruitment policies should be followed to find the right person. Thus team will work as Agile team.

• Start pair programming in the team because by pair programming as two developers sit together for coding, so knowledge and programming skills are shared.

• Start involving client at every stage to get earlier feedback.

3.4 THE BENEFITS

The proposed mapping function can be well-designed and purposeful when all the parameters are identified in the existing traditional model of the organization and transformation is done by mapping according to the mapping parameters. The major benefits from this mapping function are as below:

• Time consumption would be less because to apply mapping function is very simple.

• Everybody would be happy (team, customers, top management) as the project will be delivered on time and within budget.

• Old resources of the organization would not be unemployed.
3.5 CONCLUSION

An Agile model is proposed for adopting Agile processes in the software industry. A mapping function is also presented for transformation from traditional software development model to new Agile model. It is the base to implement Agile and victory rate of any Agile project can be increased by matching all the parameters of the mapping function.

Once the environment for Agile has been set up, the proper estimation in Agile can be done. The next chapter proposes the estimation techniques.