Implementing Agile for Enterprise Adoption

You have traditional approaches to software development and project management like Waterfall which are plan-driven. Then you have Agile where the focus is on generating early releases of working software using more collaborative techniques and frequent customer involvement. Is one better than the other or more successful than the other? That likely depends more on the type of organization you are developing software in and the type of customer and industry you are managing projects for. However, the general consensus is that Agile methodologies are an efficient way of producing software with significant advantages in production costs, time-to-market, complexity, and quality improvement over heavy-weight traditional methodologies such as Waterfall.

Even with the research showing advantages of the Agile software development practices and methodology, there is still resistance. Any change brings challenges, costs and extended timelines – not all organizations are ready to take on these challenges and not all user bases are ready to make
major changes to their long established way of doing things and performing their jobs. Change is painful for many, not to mention scary, confusing, frustrating and carries the always apparent potential for failure and embarrassment.

Introducing Agile practices into an organization can be a challenging task. For new adopters the case for Agile methods has to be supplemented by results showing the high failure rates in traditional projects. Highlighting the inherent defects in the Waterfall model and promoting the idea of iterative development could also serve as useful strategies. Adopters of Agile methods usually face difficulties in assigning new roles to the management, bridging the move from legacy code to Agile, and negotiating with leadership on the issue of transformation.

Why chose Agile?

Agile software development methodologies have attracted considerable attention from the information technology IT industry in the past 10-15 years. The reason for this interest is usually attributed to the following advantages over traditional plan driven approaches...

Improved ROI (return on investment). The usual scenario has customers providing frequent feedback on each iteration, allowing the development team to create better software and improve the ROI. It’s not a perfect scenario, but in many cases – especially when the work or the scope or the requirements has the potential to move and change, Agile can usually provide the better ROI since it can flex better and help eliminate the re-work that would be involved with moving requirements in the Waterfall methodology scenario.

Early detection of failing projects. In traditional project management and software development, optimistic reporting on abstract tasks such as analysis and design, delays problems from being identified in the early stages of the project. Agile methods execute design, analysis, and implementation tasks repeatedly in short iterations allowing greater visibility on the state of progress of the project. Sponsors can cancel the project early if they find it is not going as expected – minimizing their financial exposure.

Increased flexibility – Agile methodologies are a great response to changing or less defined requirements. Agile practices place an emphasis on creating cohesive teams of developers that collectively analyze, design, and code software. This eliminates the possibility of any bottlenecks in the development process because of developers working individually on tasks.

Increased control over projects. This one can be debated. However, Agile processes focus on people over processes. With less stress on documentation and more attention on delivering functionality at the end of each iteration, Agile teams can improve their speed to go-live with time-sensitive, needed functionality. Short iterations, multi-disciplinary teams, knowledge sharing, continuous integration, and feedback allow better control over the project and increased visibility.
Higher software quality. Test-driven development, short iterations, scoping, and frequent customer feedback have been found to generally contribute to the improved overall quality of the software solution.

Enterprise resistance to adoption of the Agile way

Close-minded user community. There are many software professionals who have not invested time and energy to learn about upcoming and promising methodologies. These people belong to two camps; one group that perceives Agile methods as nothing but break / fix coding in disguise, and the other group that has adopted an anti-Agile stance. It is extremely important to actively educate these people about the advantages of Agile methodologies. Some people would intentionally spread incorrect information about agility to prevent Agile introduction into the organization. Such misinformation should be challenged and prevented from being spread. Unfortunately, a small number may dig their heels in and completely resist change. And again, unfortunately, these individuals may need to be shown the door if the enterprise is truly committed to the change.

Fear of change. Doubts in people’s minds that they would not be able to acquire the required Agile skills or cope with the new Agile environment may instill fear in their minds. Since fear of change is usually associated with a sense of loss, these individuals may protest Agile introduction and try to undermine conversion efforts.

Serial thinking. Many IT professionals have become accustomed to serial approaches and are un receptive to new and evolutionary approaches. This can be attributed to the fact that much of the past 40 years have been dominated by software development methodologies using serial approaches. These people want to identify the complete requirements first, then design the system, and only after that start coding. Such people need to be given appropriate training, enough time, and targeted mentoring to learn the principles of Agile development. At the same time, the organization must be careful that the overall serial mindset does not kill the enterprise rollout of Agile practices.

Wrong skill set / specialized skills. Thirty plus years of serial development has resulted in people being trained in specialized areas of software engineering. This causes professionals to have highly specialized skills in a particular area while having minimal or no knowledge in other aspects of development. For example, project managers without an understanding of the underlying technologies used by their teams, programmers with no analysis and design modeling skills cannot be effective in delivering high quality software. To solve this problem professionals should train to become generalizing specialists so that they have specialized skills in one or more areas as well as basic understanding of the technical and business aspects of software development. The advantage of comprising teams of generalizing specialists is that everyone can use artifacts that are easily understood by all team members.
Outdated skills. Many people face a steep learning curve because they have not brushed up on their skills for several years. These people are also unlikely to be up-to-date about the latest happenings in the development of new technologies and processes. Cuts in education and training budgets by enterprises can contribute further to Agile adoption problems. People should be given enough time and training to improve their skill set for the benefit of their careers as well as the organization as a whole.

Black and white thinking. The black and white mind-set is prevalent among many IT professionals. People consider that they have to choose from a limited set of options and that there is no scope of selecting a middle path. For example, most large organizations will recognize the fact that a combination of Agile and traditional practices is best suited for them over complete adoption of Agile methods. Inflexible people could make Agile introduction difficult. Education and training can serve as the most suitable counter-active measure for these individuals.

Documentation-heavy mind-set. Many professionals have the wrong notion about developing effective software as being based on producing very detailed and comprehensive documents, plans and requirements. As mentioned earlier, Agile methods focus on code development over document creation and detailed requirements as this process is better suited to a more flexible development environment and plan. Project and development resources will need to be educated in the Agile approach to documentation and retrained to adopt the ‘document as needed’ method.

Do-it-all-at-once attitude. It is a serious mistake trying to adopt all Agile practices all at once – force feeding it on the enterprise population. To make the evolution of the work environment and the transition to Agile practices smoother, Agile techniques should be implemented one at a time and over a reasonable time period. Full adoption, cooperation and retention of onboard skill sets and dev/project resources should be the ultimate goal whenever possible.

Politics. Office politics is an essential part of everyday life in enterprises. Some people will prevent the use of Agile practices to save their power base. For example, a senior programmer who has a corner office may not like sharing space with junior programmers in a cubicle. A group of politically savvy professionals may at first accept agility. They would, however, wait for Agile practices to fail and revert to old practices. One should become politically adept for achieving success in introducing Agile methods.

Concepts to smooth the transition

Start small. Enterprises should use pilot projects to prove to themselves that Agile methods work for them, use pathfinder projects to determine how to integrate Agile processes into existing enterprise processes, and then move to production projects. By carefully selecting projects as try-outs for the Agile methodology, one can determine the appropriate project for the pilot. The pilot project should have committed business sponsors to show that the project had good sponsorship, deliver
high business value to show that Agile practices can produce high-quality working software in short amount of time, and be of short duration to see quick results.

**Senior leadership backing.** The upper and middle management in the organization should be educated about the benefits of Agile approaches. In most organizations, a direct move to Agile practices could be difficult. It is more advisable to initially promote Agile practices more slowly or as recovery tactics for failing projects. The buy-in process for such projects would be easier as the desire to rescue the project and try a new approach is high. Managers faced with the dilemma that by using Agile methods they would not be able to make product commitments to customers should be told that any estimates about cost, time, and functionality of previous projects were probably wrong, heavily padded, or both. After convincing management about their history of incorrect estimates, they would be more susceptible to trying Agile methods. If an organization has a record of on-time delivery of software, then the management should be advised that Agile practices could result in early completion, lesser consumption of resources, or both. Managers concerned with progress tracking issues in Agile projects could be given model status reports like burn graphs, percentage of tests passed, commentary on project state etc. based on fictional data of an enterprise using Agile processes.

In some cases, managers are concerned that Agile projects going on forever, because iterations will continue as customers keep on identifying high-priority tasks. To solve this problem, the management could be convinced to sponsor a project for a certain period of time based on estimates of when the project would be completed and any additional funding could be provided after discussion at the end of that period.

**Education and support of the end user community.** The management and the development team should be familiar with the principles, practices, and framework of Agile methods before actually implementing them. Training and support should be provided to eliminate any misconceptions and tackle fear of change. People should also be taught that Agile development is a highly disciplined approach and a half-hearted approach would not produce the desired results. Overenthusiastic teams should understand that decisions made without foresight will lead to eminent project failure. Empowered teams are an essential feature of Agile methods. Responsibilities associated with empowered teams should be gradually communicated to developers. Because many Agile practices require daily stand-up meetings, people consider Agile methods as professing micro-management. To avoid this concern, managers should be ready to answer any questions posed by developers and remove obstacles promptly. Managers should also not complain if some tasks take longer than scheduled.

**Thoughtful integration with external processes.** It is important that Agile practices are tailored to integrate with existing non-Agile practices in the enterprise. This means being ready to produce lower-value deliverables and creating plans or status reports in traditional formats for obtaining
the necessary approval. Once the benefits of Agile processes are clearly visible, negotiations on the balance of Agile and non-Agile approaches can be started. When introducing an Agile process, the management should understand the impact on groups outside the development group. The performance of the development team could be negatively impacted by other groups using non-Agile processes. Managers should find ways of resolving differences of opinions among these teams. The human resource (HR) department should be informed about any development team moving to Agile practices. The HR department could share concerns on the imprecise deliverables and dynamic scope of the project. The HR department and the team could work proactively to define tasks and deadlines that satisfy the HR department’s need of deterministic action plans and also allow the project to remain Agile.

**Enterprise status updates and rollout.** The success of the pilot project could be showcased to attract more sponsorship for adopting Agile practices. Management should raise awareness and remain open to feedback. They should use the experiences of the team involved in the pilot project to plan for future pathfinder projects. This team could now act as mentors for the teams working on subsequent Agile projects.

**Sustain the growth.** Having adopted Agile processes, it is necessary to sustain the Agile work environment. Constant encouragement and stimulation should be provided to prevent people from going back to their old ways over time. An atmosphere of discussion should be promoted through special interest groups, conferences, practice sessions, etc. to keep the enterprise adoption effectively moving forward. Avoid an adoption slow down or halt at all cost – it could be a momentum killer.

**Summary / call for input**

The bottom line is this – Agile isn’t for everyone, everywhere and every organization. But it has shown, over time, to be the right move for most who have taken on the challenge. Starting small with a pilot project and expanding that adoption window is the right way to enable the organization to quickly revert to the old ways if needed. However, full buy-in and commitment and transition should be the plan – otherwise why try at all.

Readers – what are your thoughts on Agile adoption? Has your organization taken on the move to Agile methods? What walls did they come up against? How did they respond – how did the resistant workforce respond? Who did project success change?