BUSINESS EXCELLENCE

Agile in the IT World

kpmg.com/in
Abstract

When it comes to process, Agile is the buzz in today’s IT world. What exactly is an agile process and how different it is from the traditional waterfall model has been illustrated? This has been done by portraying the challenges that an IT industry faces. The theory behind agile practices, and its values and principles have been discussed elaborately. This is where the concept of a 2-pizza team size comes in; that is a team should be just enough the size that 2-pizzas could satisfy its appetite. These concepts drill deeper into benefits of using agile as a methodology in an IT environment and how costs can be reduced while bringing in the ROI early. A model has been introduced for IT process control demonstrating various factors that come into play. Recent trends on adopting agile as a process and respective success rates have been detailed out from a research. There is also a summary of few of the most popular agile methodologies, along with discussion on the metrics that can be used in an agile world. These discussions provide a motivation for researching into the agile methodology and discovering new avenues of growth using the same in an IT world.
Introduction

In today’s dynamic age, businesses change rapidly and act to competition, hence requirements tend to change within projects. In such scenarios, costs of change can be overbearing and there has to be a way to flattening those costs. It’s not just difficult for the developers, but even unnatural for the customers to envisage the fully implemented product without seeing it evolve. It’s almost like drawing with a blind-fold without being able to see how the drawing evolves.

Agile practices are a way to diminishing those costs and displaying quicker results with lesser ceremonies. Here, the paper introduces agile methods for product development in the IT world as an alternative to the traditional waterfall model. It first compares the traditional model with agile practices, and then awakens into the agile world with its concepts and principles. It then also analyzes how agile helps redefine success to teams, especially in a software environment, where dynamism is prevalent and adaptability is key.
Challenges in the IT Industry

Today, the IT industry has a disastrous track record where most of the projects fail with respect to on-time and within budget delivery. These failures can be attributed to few factors as below.

- **Change in Requirements**: There are numerous assumptions made from the inception till the closure of any project, which are the biggest contributors to surprises by the time product is realized. One of the major ones being “we know the requirements”. Case studies show that no requirement can be carved in stone. Requirements tend to evolve and hence eventually might change too. Tackling with these evolving requirements would lead any project towards success.

- **Lack of Stakeholder Involvement**: In a traditional environment, there is an evident lack of stakeholder involvement. There is no communication channel between the customer and the developer and no continuous collaboration between the developers and their managers. Such factors signify a potential pitfall since the customer lacks visibility into the product being built. Bringing users and developers together is a great way to effectively address business requirements.

- **Early Product Realization Issues**: In a traditional development environment, stakeholders including developers wait till the final phases of the project to see a working product. This tradition injects defects along the way and accumulates them to be tested only until the product is delivered.

- **Unrealistic schedules and Inadequate testing**: In an IT environment where hierarchies are involved, estimations are barely a collaboration task. Schedule estimates are assigned and milestones are decided by the upper management. The team is then required to adhere to that schedule. This might result in a lot of time being spent in the initial phases, and the schedule getting slipped, as a result of which testing might not be done in its entire range and the product eventually delivered with defects.

- **Process as an Overhead**: Generally, processes in an IT environment are so inflexible and bloated up, that they only contribute to overhead in the team instead of providing best practices that can be implemented as suited to the team’s environment.

- **Wastage of Features**: It has been noted that only a small percent of the features developed as part of a product is used always. A huge portion of the product functions are never used. The below figure gives the exact numbers, and offers a different perspective to limitations that a waterfall approach suffers from. This can again be attributed to above two points on lack of stakeholder involvement and early product realization.

![Features and Function Usage in IT Industry](http://www.ibm.com/developerworks)

- **Cost of Change**: In a traditional waterfall model cost of change increases exponentially as we move along the different phases in the lifecycle. This is because, the waterfall model requires all decisions to be taken at the inception of the project, and the later the changes occur in the lifecycle, the harder it is to incorporate them as it affects all the deliverables behind it.
The waterfall model

The waterfall model is one of the foremost methodologies used for software development and is also prevalent in many industries including services and manufacturing. Although this model has a proven success in terms of efficiency and a detailed coverage of the product lifecycle with sufficient monitoring and control, it has several limitations too.

Firstly, it enforces all the requirements to be made available to the designers even before the design phase starts, and similarly the entire design being in shape before coding could start. This does not reflect reality in the IT world. Unfortunately, software cannot be put on a conveyor belt where all processes and implementation methods can be predetermined, and the products would be manufactured just-in-time with zero-order defects.

The pith of the situation is that there’s a steep fall in the motivation of the stakeholders as we move from requirements to the delivery. There is nothing to back the motivation of the team members and keep it going with a constant pace. The product is not tangible until all the implementation and testing is complete, and hence the customer can do nothing but just wait until the product arrives with surprises. So who is to be blamed – the customer for not being specific about the requirements and requesting changes, the developers for not understanding the customer’s view of the requirements and struggling with change management, or the process itself that’s so bizarre that it cannot adapt itself to realities?

We discuss precisely the challenges that about 13 percent\(^1\) of the IT industry faces for the fear of paradigm shift from the waterfall methodology.

---

\(^1\) “Agile Adoption Rates Survey Results” by Scott Ambler at http://www.ambysoft.com/surveys/agileFebruary2008.html
What is Agile?

So what exactly is “Agile”? It’s an umbrella constituting various incremental and iterative software development methodologies, where requirements evolve and are developed by continuous collaboration across teams throughout the lifecycle of the project. All iterations are time-boxed and there’s barely any scope for slippages.

It’s essentially the values and principles of an agile process that make it completely agile. An agile process believes in adaptability, transparency, simplicity and unity and maintains these values using best practices like continuous collaboration and embracing changes in requirement. It believes in the principle of keeping it simple and just enough process is exercised by a set of self-motivated and self-organizing individuals in the team. The agile manifesto and the related concepts are discussed in the sections below:

The Agile process

The Agile Manifesto

In February 2001, 17 developers met at a resort in Utah to discuss the challenges they were facing in their respective development environments. This discussion eventually led to effectively a charter known as “The Agile Manifesto.” Agile Manifesto lists a set of values that if believed in and implemented would result in a shift from traditional development methodologies to being more flexible and realistic in the IT world. It reads as follows:

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

2 “Agile Software Development” at http://agilesoftwaredevelopment.com/
Agile Principles

“Agilists” in their organizations abide by the following principles which underlie the agile manifesto, and without which no process can be termed agile.

• First and foremost: Satisfy the customer - Deliver working, valuable software early and frequently
• Measure progress primarily by working software
• Have business people and developers work together daily
• Welcome changing requirements
• Create a self-organizing team of motivated individuals
• Communicate using face-to-face conversation
• Avoid nonessential work
• Maintain a sustainable pace of development
• Attend continuously to good design
• Retrospect and adjust regularly.

The Concept of 2-Pizza Team Size

“If you can’t feed a team with two pizzas, it’s too large. That limits a task force to five to seven people, depending on their appetites” - Ken Schwaber.

According to Ken Schwaber, one of the co-authors of Scrum, an ideal team size for an agile process is 7+/-2 as per the 2-pizza team size concept. Ken thinks if you cannot feed your team with just 2 Pizzas, then you are falling prey to the problems of less accountability and social loafing.

According to a 2005 study on social impact theory, “as group size and dispersion grew, the group’s work would be affected in the following areas: Members will contribute less in both quantity and quality, final group output will be of lower quality, and a group’s output will be affected both by individual factors and contextual factors.” Hence, staying close to this concept aids in abiding by the agile principles.

Why Agile? - Agile comes to Rescue

Essentially, agile practices overcome the common challenges in an IT project via below listed characteristics:

• Validated product early: After every iteration, the increment is demonstrated to the customer. Since parts of the product are visible to the customer regularly, s/he is quite capable to visualize the product in its entirety, analyze risks involved along with the team, and validate the product each iteration. This inhibits defect accumulation with regard to requirements, design, and code

• Early ROI and lower costs: Costs of resources used in extensive planning in the initial changes and then costs of rework are reduced. In addition, due to prioritization of requirements, high-valued features are implemented first and hence higher return on investment is realized in the early stages

• Embracing Change: Agile principles are based on adapting to evolving requirements from the customer. As the customer sees his/her product being built, he/she has a better view of the product, and hence requirements get refined along the way. Agile values this and costs involved with such changes are much lower here. Barry Boehm had indicated an exponential increase in cost of change as we move forward in the lifecycle. Kent Beck, creator of XP and TDD, created his own model of cost of change that challenged Boehm’s curve. It espoused that change can be inexpensive even late in the lifecycle while maintaining system quality. Cockburn and Ambler revised this “flat” curve to reflect modern corporate realities.

• Customer and Developer Satisfaction: Since agile practices believe in continuous integration and feedback loop, the customer gets to validate the product inch by inch. Hence there are lesser defects and lesser surprises left for the end, resulting in a satisfied customer. At the same time, the developers do not have to follow bloated processes that do not suit their way of work, and inhibits any control from the upper management. This gives agile an edge of having satisfied team members too

• Scope for Innovation: Agile practices encourage initiative, innovation and teamwork for all kinds of activities. Also the must-have features get built first, leaving the nice to have features for later iterations, thus facilitating creativity to the product and resulting in scalable projects.

3  “Agile Principles” at www.agilealliance.org/principles.html

© 2011 KPMG, an Indian Partnership and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative (“KPMG International”), a Swiss entity. All rights reserved.
Factors Influencing Agile Processes

Almost every IT project has evolving requirements and risks involved across all phases of the lifecycle. Thus, no process can be defined in pen and paper to handle any situation that might arise. This practical problem led to an empirical approach to project management where processes could be adapted periodically as per the need. Scrum is one such process which follows empirical and adaptive project management approaches to continuously revisit the plan and re-factor it.

Any project management approach needs to closely analyze the four key variables that affect process control. These are:

- **Cost (or Effort)** – Budget guides the amount of effort that can be expended
- **Schedule** – As timelines change, the project is heavily impacted
- **Scope** – Any scope creep influences project management activities downstream
- **Quality** – Quality could be compromised depending on cost, schedule and scope factors hence affecting overall process control

If we were to put it mathematically,

\[
\text{Process Control} = f(\text{Cost, Schedule, Scope, Quality}).
\]

where, Cost, Schedule, Scope & Quality are interdependent.

---

**Cost of Change Patterns by Boehm, Cockburn/Ambler and Beck**

Source: Reexamining the Cost of Change Curve, by Alistair Cockburn; XP Magazine, September 2000.

---

**Factors Influencing Control over an IT Process**

- **Skills**
- **Quality**
- **Competency**
- **Culture**
- **Schedule**
- **Scope**
- **Development**
- **IT PROCESS CONTROL**

---

In agile environment, people factors play a big role in addition to the 4 variables. Agile requires continuous collaboration between team members and a self-managing team where people skills can affect control over the process, and eventually successful adoption of agile. Hence, additional factors that need to be considered into this equation are:

- **Skills** - Team skills can be an important attribute while selecting best-fit process and defining an empirical approach for the project.

- **Competency** - When a team needs to motivate and organize itself, competencies play a major role.

- **Development** - There can be learning curves involved in view of varied technologies used, and hence this needs to be accounted for too.

- **Culture** - If members in a team trust each other, it creates positive environment towards making the project a success.

**Recent Agile Adoption and Success Rates**

The figure below shows the recent trends in agile adoption and how successful these adoptions have been.

**Cost of Change Patterns by Boehm, Cockburn/Ambler and Beck**

![Diagram showing adoption and success rates](source: Agile Adoption Rates Survey Results by Scott Ambler at http://www.ambysoft.com/surveys/agileFebruary2008.html)

© 2011 KPMG, an Indian Partnership and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative (“KPMG International”), a Swiss entity. All rights reserved.
As per the figure, more than half of the IT organizations across the world are already following agile in some way or the other. About 24 percent have gained credibility in agile practices and would be going the agile way. Of the 69 percent who are practicing agile, success percentages are very high and only a quarter of them show success rates of below 75 percent.

Below charts are results of qualitative surveys taken across IT organizations (mostly software development) that exercise agility. It gives details on how productivity, quality, stakeholder satisfaction and costs differ from a traditional model. As is evident from these numbers, all of these variables show satisfactory results.

Productivity

- Much Lower
- Somewhat Lower
- No Change
- Somewhat Higher
- Much Higher

Quality

- Much Lower
- Somewhat Lower
- No Change
- Somewhat Higher
- Much Higher

Business Stakeholder Satisfaction

- Much Lower
- Somewhat Lower
- No Change
- Somewhat Higher
- Much Higher

Cost of System Development

- Much Lower
- Somewhat Lower
- No Change
- Somewhat Higher
- Much Higher

Some popular Agile methodologies

There are multiple agile methods that are shaped by agile principles like Unified Process modeling, Dynamic Systems Development Method (DSDM), Adaptive Software Development, Crystal, Feature-driven development (FDD), etc. A brief summary of these follows.

**SCRUM** - Scrum is one of the agile methodologies that most of the fortune 500 companies have successfully adopted. Scrum revolves around its iterations known as sprints which are typically, 2-4 week cycles. Initially, there is a product backlog which lists down all product requirements in form of user stories and a sprint backlog which consists of selected user stories for that sprint. At the end of each sprint, we have a shippable item that can be demonstrated to the customer.

In essence, Scrum has 3 roles- Product Owner who is the voice of the customer and creates the product backlog, prioritizes items in the backlog, and ensures that the team delivers value to the customer’s business, ScrumMaster who facilitates Scrum for the team and removes impediments for the team, and the Team that is made up of 5-9 people with cross functional skills who do the work and is self-led.

**XP (Extreme Programming)** - XP is another agile methodology that revolves around small development cycles in order to increase developer focus and productivity. It’s based on concepts of simplicity, communication, feedback (using unit tests and acceptance tests) and courage. As in Scrum, user stories are developed by the team members and then are converted into test scenarios. There are positive and negative test cases written for each user story and pair programming is encouraged to write the code. When requirements change from the plan, the developers embrace this change and plan this requirement in the next release. Figure on the next page depicts a typical XP process.

In addition, Scrum focuses on the following meetings:

- **Daily Scrum Meeting** - This daily discussion of 15 minutes focuses on what team members accomplished yesterday, what they plan to do today and any roadblocks in their way.

- **Sprint Planning Meeting** - It occurs before every sprint in order to prepare the sprint backlog and assign tasks to team members.

- **Sprint Review and Sprint Retrospective Meeting** - At the end of each sprint, it focuses on reviewing the tasks that were completed in the last sprint and demonstrating them to the customer. Further, there’s also a retrospective meeting where the team discusses lessons learned (both positive and negative) and improvements needed in the upcoming sprints.
Dynamic System Development Method: DSDM was developed in order to remedy IT projects that run on tight schedules and budgets and lack stakeholder involvement. There are about 8 principles underlying the DSDM approach which include focus on the business need, iterative development and no compromise on quality to name a few. The activities in a DSDM methodology span across the pre-project phase, the project life-cycle phase, and the post-project phase. The project lifecycle phase has 4 major steps- Study, Functional Model Iteration, Design and Build Iteration, and Implementation. DSDM was originally based on the Rapid Application Development (RAD) methodology and is administered by the DSDM consortium.

Feature Driven Development: FDD is another of the agile methodologies, driven by features (client-valued functionality) and timeboxed to short iterations. As with other agile methodologies, its main objective is to deliver working software (features of the product) every iteration in a timely manner. It consists of five basic activities- Develop overall model, build feature list, plan by feature, design by feature, and build by feature. These activities are tracked using completion percentages on each milestone and are implemented through best practices like inspections, regular builds and the like.9

9 "Feature Driven Development” at http://www.featuredrivendevelopment.com/
Measuring team success using Agile metrics

The objective of an agile process is to make problems visible to the team and stakeholders, so that measures can be taken to overcome the roadblocks and continuously improve the process as it suits the team. Metrics discussed below make the performance of the team transparent, indicating the dos and don’ts and hence eventually driving them towards success. There can be multiple attributes to each of these metrics, and they provide a whole new perspective to measuring work, progress and delivered value.

- **Burndown and Burnup Charts** - Burndown charts are essentially progress charts that give direct measures of work remaining over a lifecycle, impact of scope changes and projected completion dates. This burndown can be measured in terms of number of releases/user stories/hours of work remaining. It’s a good visual that reflects how the team is performing on the project, and deviations if any (from the plan/ideal) can be analyzed for process improvements. It’s a good tool to indicate causes of effort and schedule variances. Burnup charts on the contrary indicate the amount of work that has been completed by the team.

- **Velocity** - An agile team’s velocity measures the rate at which the team is implementing the requirements or user stories and gives an insight into productivity and team efficiency. Faster the team completes a user story, faster it can be demonstrated to the customer. Hence velocity is directly related to customer satisfaction. Velocity is a clear reflection on a team’s commitment to develop user stories and their estimations, and is comparable across iterations in the same project.

- **Earned Business Value** - This is one of the most crucial metrics in an agile environment where satisfying the customer is key. Since agile promises early ROI, this metric is a direct measure of value delivered to the customer. According to Dave Nicolette, “Earned Business Value (EBV) may be measured in terms of hard financial value based on the anticipated return on investment prorated to each feature or User Story.”

- **Running Tested Features (RTF)** - RTF is a direct measure across iterations of the number of tested features that have been implemented and are in use. Since working product is a primary measure of progress (and ultimately success) in any agile project, RTF gives an angle to look into increments that have already been delivered and if they are working as desired.

Above metrics are just samples of what most agile projects would use. Any agile process does not mandate use of one set of metrics over the other. It is solely dependent on the project team to identify measures and metrics that suit them best.

Limitations of Agile

In addition to the advantages of agility, there are several disadvantages that are brought along. Although Agile allows embracing changes and being flexible, sometimes too much flexibility might mean failure. If the change management process is not very strong, then the team might end up accepting all changes as they’re requested throughout the project lifecycle. This increased flexibility also gives leeway to limiting amount of documentation that is required out of any project in order to maintain the product/application. Also, since the focus here is on continuous collaboration, too many meetings within the team might be onerous.

Due to its adaptive model, agile lacks predictability that is prevalent in other waterfall models. Although process areas up to level 3 in a CMMI framework can still be mapped to agile process concepts, it is difficult to map them to the high maturity areas. Agile does not talk of a prediction model, and only believes in continuous improvement through retrospectives and other innovations each iteration.

Besides these, one of the major challenges of agile is implementing it in a distributed environment within larger teams. Agile works best when teams are co-located since continuous collaboration is expected. Also, managing stakeholder expectations might be difficult when everyone is not working from the same geographical space. Few teams have been successful in leveraging SCRUM to a team of about 200 people, but there has to be further approaches taken to successfully manage large agile teams.
Agile, with an expanding and self-sustaining pace, has reached out to nearly every nook and corner of the world where IT lives. It is no more considered as an alternative to IT process, but a must-have to break the ground rules of traditional development models. Since no process is without flaws; agile too has its own negatives of being highly successful only in co-located small teams and performing weakly in distributed environments. However, the advantages of going the agile way may outweigh the cons that can definitely be overcome using strategic development practices.

An empirical approach like Scrum goes a long way in adopting practical approaches to the IT world where a prediction model falls short. Embracing changes and encouraging flexibility gives direct gains into increasing customer satisfaction and displaying value-added results along with motivating the team. Leaving behind the tradition of assigning work and thus creating a culture of accountability are key elements in this paradigm shift. The collaborative and flexible environment that agile advocates cuts across all variables that impact an IT process control. Nevertheless, “underpromise and overdeliver” is what eventually helps teams rise to success, and agile is definitely one of the roads to it in the IT world.

Conclusion
Reviewers’ Profile

KK Raman
Partner
Business Excellence
KPMG in India

He handles Strategic Advisory Services encompassing Model based, Six Sigma, Agile and Sourcing Process Improvement Advisory Services. An MBA from IIM(B), Raman has directed over 200 advisory engagements and served on the review panels and pilot initiatives for new models and frameworks from the SEI. He is also an executive member of SPIN BLR chapter, member of European SEPG review panels, member of Institute of Public Auditors and is active on NASSCOM, CII panels and conferences in the field. KK serves as a IT Industry Champion for KPMG in India as an Advisory Partner. He has guided organizations on the application of Agile methods in combination with model based frameworks like CMMI, ITIL.

Prasanth
Shanthakumaran
Associate Director
Business Excellence
KPMG in India

Prasanth has rich experience of more than 18 years in process advisory and quality assurance. Consulted clients in achieving business objectives through improved processes by facilitating implementation of industry proven best practices and process improvement models/ reference frame works (CMMI®, TMMI, Agile and ISO 9000) to bring lasting value to organizations. He has actively helped organisations in over 20 countries adopt process improvement using model and non-model based aspects. His interest include multi-model implementation using Agile as one of the key approaches.

Authors’ Profile

Arpan Jha
Consultant
Business Excellence
KPMG in India

Arpan is a Certified Scrum Master (CSM) and has in-depth knowledge of agile concepts. She carries a Masters degree in Software Engineering from Carnegie Mellon University, USA, and has been involved with the big 4’s in the management consulting domain and also has experience in the product development space. She has worked in an agile environment on multiple projects using processes like OpenUP, Scrum, etc. and is versed with agile tools like Process Dashboard, GForge, Redmine, etc. Besides, she has written many whitepapers on Software Engineering practices and published technical reports mapping various process capability models.
The information contained herein is of a general nature and is not intended to address the circumstances of any particular individual or entity. Although we endeavour to provide accurate and timely information, there can be no guarantee that such information is accurate as of the date it is received or that it will continue to be accurate in the future. No one should act on such information without appropriate professional advice after a thorough examination of the particular situation.

© 2011 KPMG, an Indian Partnership and a member firm of the KPMG network of independent member firms affiliated with KPMG International Cooperative (“KPMG International”), a Swiss entity. All rights reserved.

The KPMG name, logo and “cutting through complexity” are registered trademarks or trademarks of KPMG International.

Printed in India.