Quickscrum

AGILE FOR PROJECT MANAGEMENT

For information, address Quickscrum SARL, 47, Rue Klock, 92110 - Clichy, France

Published by QuickScrum
France/Germany/India

INDEX

Summary

The Agile framework offers reliable and productive project management processes for executing almost all kinds of projects - especially IT or software development related. While the majority may have heard about Agile and the advantages it offers, most individuals might be unaware about the actual benefits they can derive by using Agile frameworks like Scrum and Kanban. Moreover, an issue most commonly faced by entities outsourcing their projects is managing projects successfully while working with remote or distributed teams. If implemented in an effective manner, Agile can aid in reducing waste and streamlining work processes so projects can be managed and developed in a successful manner.

Agile For Project Management - An Introduction

In the recent times, as technical advancements keep being introduced by leaps and bounds – and on a consistent basis - rendering quality into a product today may or may not require great skills. However, designing and developing a product in a pre-defined timeframe and ensuring that it adheres to stringent quality standards may indeed prove to be a difficult task for most manufacturers. With pressing end user requirements, increasing levels of market competitions, and the need to keep working overheads low, it has become increasingly important for organisations to change traditional working processes and introduce new ones which can enhance business models and help to deliver business value more often, and in a more reliable manner.

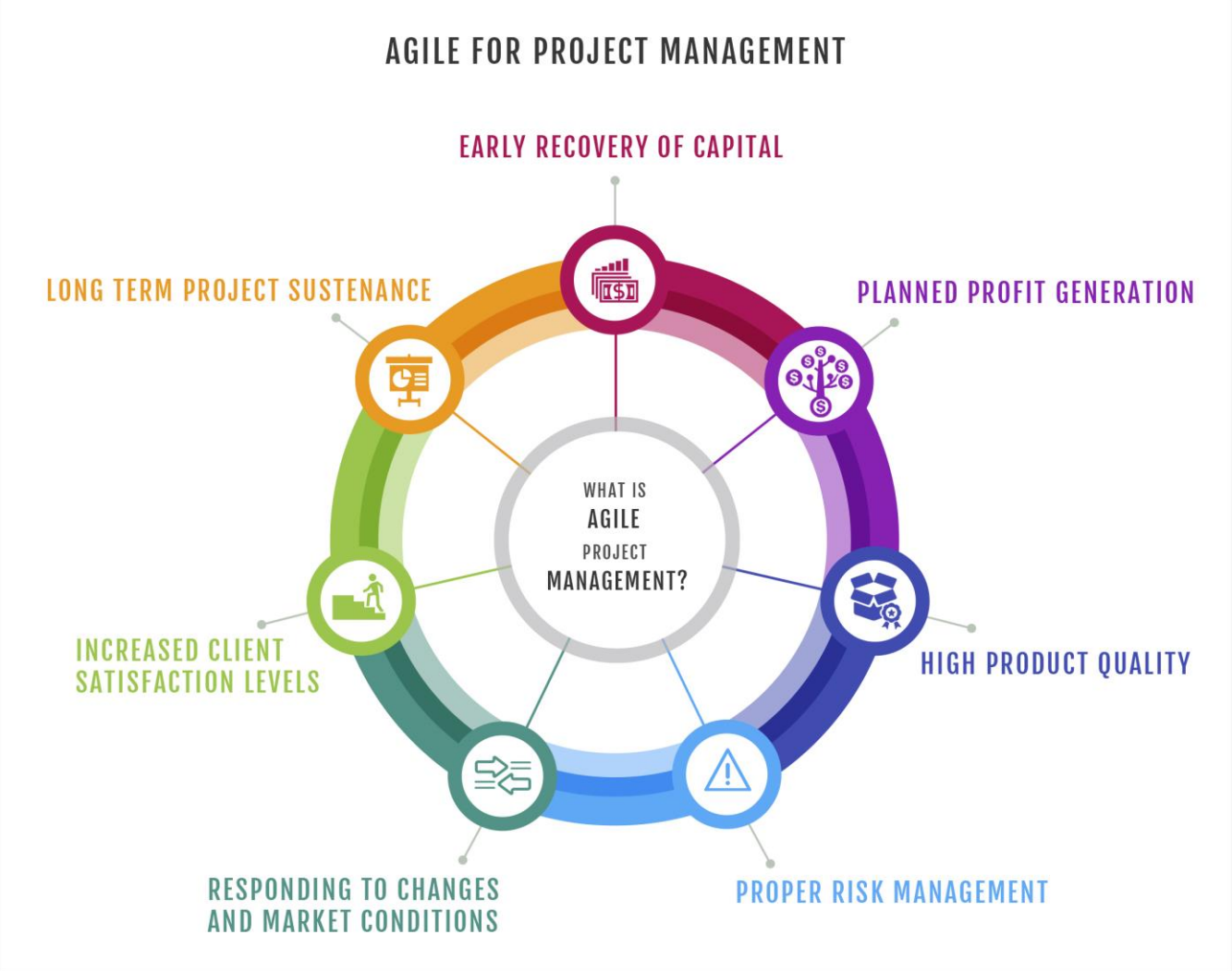
Project Management

While talking about changes, Agile tops the list of project management methodologies that support “change”. Agile was conceptualised because there was a need to find a better way of managing project such that they could be developed in less time and in a more efficient way. Perhaps the most important thought put behind the conception of Agile was to introduce empiricism into project management – something that was not done before prior to 2001 when Agile took birth in a ski resort, in the Wasatch mountains of Utah. Agile focuses upon making decisions based upon “what is” rather than “what is proposed” i.e. what best can be done with what we have currently with us. And another aspect that Agile focuses upon is to start work with very little or unclear set or requirements – you can keep on building and improving your product over time as more information is available from the client and end users. This can be a great asset to organisations as you don’t have to wait for comprehensive paperwork and documentation to get started with Agile. Moreover, you can start with an idea and keep on building it, but you need to develop it in a manner such that whatever you produce is useful and has a certain business value attached to it. So how does Agile do this? How does it work? It’s interesting to know how Agile can be effectively used to control projects that are big or small, whether they are simple or complex in nature, and if they have to be executed for a few months or for years at a stretch.

What is Agile?

Agile is a framework that focuses upon early product deliver. The products could be any - physical products, processes, software, or even running an activity or an operation. It can be understood as a set of methods and practices, stated in a manifesto, which act as guidelines, and teams have to follow those methods and practises in a proper manner to deliver the goal and objectives to somebody - usually the client. In Agile, solutions are obtained by self-organizing teams which collaborate, discuss, and constantly strive to improve their working process and become more effective in what they do. Another feature of Agile is that the teams are cross functional i.e. each team member is skilled in more than one core activity that he or she specialises in. The teams can start working even if very little information is available regarding the product to be developed. In Agile, the entire product is not delivered to the client "at one go". Rather, the product is developed in stages. After each stage is completed, the client is invited to verify that the development delivered by the team fulfils what he or she had anticipated at the time of project inception. Valuable product features are delivered by the team to the client regularly, on a consistent basis through product incremental cycles known as sprints. The sprints keep on functioning until sufficient product features are successfully developed, tested, and shipped to the client. The unique aspect about Agile is that it focuses primarily upon the delivery of business value to the client. Agile emphasises more upon how much a particular product feature is worth to the client in terms of its financial worth in the market. Important features should be developed first, followed by less important ones. When sufficient product features are developed and successfully tested, they are integrated to form a minimal viable product or an "MVP" which demonstrates the usefulness of the product and how it is likely to fare in the market. The objective of developing an MVP is to get enough inputs from the process so the management and the technical teams can decide the projects actual worth in the market, how difficult it is going to be to develop a "sellable" version having enough features so end users can use it to their benefit, how long it is likely to take, and how much capital is needed to sustain the product development process over the months.

Instead of developing the entire product and releasing it in the market after a prolonged duration, Agile concentrates upon releasing several versions of the product at regular intervals of time. Each “release” is sellable, therefore, the client can start recovering the capital invested in the project very early as compared to traditional project management methods. There are many advantages of Agile processes:



Early recovery of capital

In case of traditional models, one has to wait for the entire product to be completed before it can be shipped. The investors can subsequently start earning from the project. This can take a lot of time – often months. The investors cannot recover a single cent from the project in the meantime. In Agile, a release can be availed in as soon as a couple of months since the client can go ahead with a version with limited product features and launch it in the market. It is true that the financial worth of such a release might not be substantial, but, nonetheless it can help the client to start recovering some of the capital pumped into the project development activity. Therefore, the investors benefit from a steady trickle of cash flow till the full-fledged release is developed in the near future that can help them to earn bigger profits.

Planned profit generation

There is a certain risk involved with almost all traditional project management systems. The success levels of the product cannot be determined unless and until it is developed fully and launched in the market. As the development process takes a lot of time, there could be instances of market conditions changing in the meantime. The investors may well face a scenario wherein the conditions may have changed drastically and a competitor might have already launched a similar product. If such were the case, the bulk of the market share may be captured by competing products and investors may fail to churn out significant profits from the product. This does not necessarily happen

with Agile. Several releases can be planned over the months, with each release offering a more advanced set of product features. Managements can design a well-planned sales strategy catering to each release and target specific customers to fulfil the sales goal for each release. This also helps to mitigate several types of risk factors generally associated with large scale projects.

High product quality

One of the biggest advantages of Agile is that the testing and QA activities are done on a consistent basis after each feature is developed in a sprint. A feature can be considered as shippable only after it is tested by the QA team members and approved by the product owner. In addition, the client too reviews the features and may reject their development if he or she feels it does not fulfil the product vision. Agile processes offer quick feedback regarding the product quality. This is because all Agile frameworks are uniquely conceptualised to define and deliver the business value contained in the project.

Proper risk management

Risks are an inherent part of any project, and project management models should try to minimise their occurrence if they are to be effective. Risks keep on multiplying if they are not identified in time, and they can also lead directly-indirectly to technical debt. The basic advantage of Agile is that you don't work with the entire product at any given instance of time – you engage with a certain feature, or a set of features, and develop it during the sprint. It is easy to test for and identify risks when individual features are checked for their reliability and this is what Agile proposes to do. Agile teams are cross-functional i.e. a developer or a programmer may double as a QA tester, while, certain teams may include professional testers who carry out unit tests, manual tests, Selenium tests, and even use automated testing software to check for redundancies and bugs in the software code. Regression is checked for at the time of feature development, and if a particular feature fails to perform as per the acceptance criteria stated in the QA conditions of the feature, the said feature is rejected at the time of review and transferred back to the backlog for redevelopment. Another positive aspect about Agile is that its principles categorically state to avoid rework which can leads to waste, or money and efforts spent over activity that consumes time and does not produce any positive results.

Responding to changes and market conditions

Project development takes time. Large projects may take months before they can be deployed in the market. Considering the dynamically changing market trends, it is quite possible that some product features may become outdated by the time the project is launched. This is a typical scenario in case of projects using traditional "staged" project management methods in which product features or their scope cannot be changed once finalised. Waterfall methods are irreversible, therefore, it becomes very difficult to incorporate changes in the product design once the documentation is done. this limitation is removed since the framework is specifically envisioned to respond dynamically to changes occurring in the product design induced due to changing market trends. The main advantage you have is you develop a small part of the project at a time in incremental cycles known as sprints. If a product feature is developed and it is later realised that it does not totally fulfil end-user requirements, or if the requirements have changed over time, using Agile it is possible to update the particular feature by taking it up in a sprint and redeveloping it as per new requirements and acceptance criterions.

Increased client satisfaction levels

Client satisfaction depends primarily upon how much he/she gains from the project, and in terms of project development, how a project will fare in the market and earn high ROIs. Apart from the financial aspect, another factor which contributes towards increased client satisfaction levels is up to what extent the management or the client feels happy with the development activity. Often clients have product visions which teams fulfil partially or fail to satisfy. A primary reason why this happens is the client fails to convey a proper vision to the team, or the team fails to understand it properly. The teams may offer productivity which the client does not actually desire but realise the fact only when the project is deployed. This is because traditional management methods do not encourage client participation to a great extent. In Agile, each product feature can be considered as “Done” only when it is approved by the client. The client is very closely involved in the product review activities, and the team may have to do rework on a particular feature if the client is not satisfied with a particular feature development. The client feels very much in control because he or she is closely involved with the project and remains updated regarding recent developments.

Long term project sustenance

Market trends indicate end users having specific requirements are not ready to compromise upon, or wait for indefinite periods of time to fulfil their work related needs, primarily because they have other options available in terms of newly emerging brands and start-up ventures. Competition is rising, and will continue to do so in the near future as new development technologies keep on being introduced in the market. Vendors don’t have many choices but to develop products that focus totally upon end user requirements and satisfy them. This means whatever products they develop have to include a plethora of features and offer them at competitive rates to beat the competition. This also means development will take time and projects will extend over months before a MVP is developed, tested, and modified to produce a saleable version. It is easy to scale Agile by adding or removing team members depending upon the project complexity and product related requirements. Managements can control the production costs by investing or scaling down development resources since the development activity is carried out in sprints in Agile framework, and Agile supports such changes. You can speed up the release date, as per needs, or postpone it to a later date if new features are to be added in the production plan. It is very difficult to do this using Waterfall methods since the scope of the project is predefined and resources cannot be scaled up or down. Moreover, if the project is rescheduled to extend for more months, it could run out of funds and employers may not be able to support the project any longer.

Distributed Teams Challenges and Agile Advantages

Project management methods and Agile

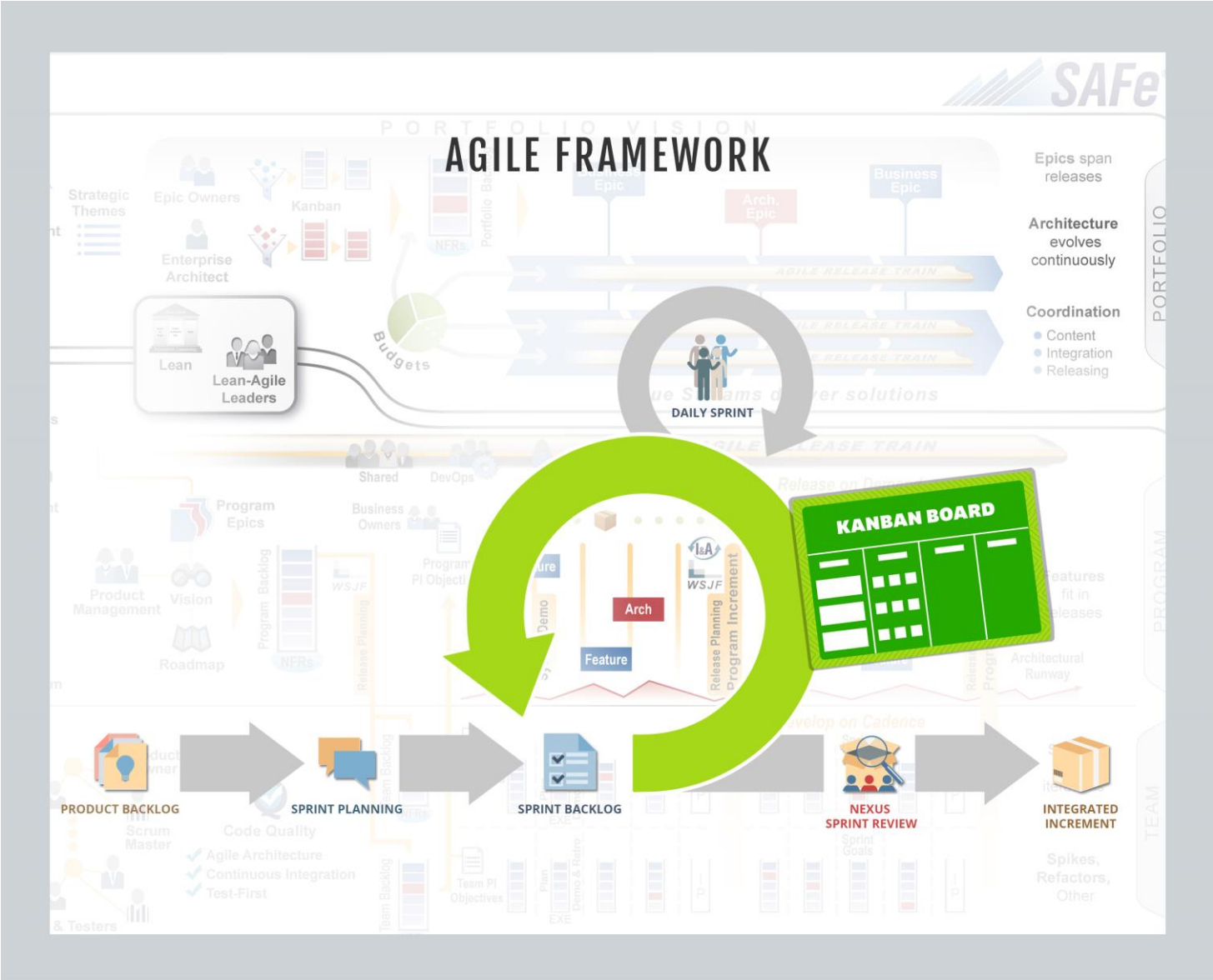
With dynamically changing market scenarios dominating the outsourcing markets, it has become imperative to remain conversant with emergent technologies and use them for developing projects. New platforms and technologies have a lot to offer in terms of reduced development time and targeting a wider range of client centric requirements, however, while reaping the benefits they offer, they also impose a few constraints regarding their applicability. Offshoring businesses can increase the productivity levels and generate higher profits but often face problems in finding technical teams familiar with the usage and implementation of new technologies. For most organisations, it is more profitable to find technical talent in other countries and outsource their projects depending upon the nature and scope of the project on hand. It is very important to manage projects in an effective manner to make them profitable. Several project management frameworks and methods aim to make project management easier and more effective. Some of the popular methods used in the past, and even now are:

- Critical Path Method (CPM)
- Critical Chain Project Management (CCPM)
- PMI/PMBOK Method
- Event Chain Methodology (ECM)
- Extreme Project Management (XPM)
- Adaptive Project Framework (APF)
- Lean Development (LD)
- Six Sigma/Lean Six Sigma
- PRINCE2
- Dynamic Systems Development Model (DSDM)
- Feature Driven Development (FDD)
- Rapid Application Development (RAD)
- Systems Development Life Cycle (SDLC)
- Waterfall (Traditional)

Each method proposes to make project management easy and more accurate. Often, it is difficult to choose which method one ought to adopt for developing a project since every management technique has its own pros and cons. While a particular organisation may offer a positive feedback regarding a method it is following, consultants might consider it a bad choice and speak against it. There are no postulates or rules which define a “successful” project. Also, there are no rules which can help in deciding whether a particular methodology is more effective as compared to the other. It is based more upon personal experience, understanding how a methodology works and what it has to offer, and how well it can be implemented. Perhaps, the most important aspect to understand is whatever methodology you choose, what is more important is how well you use it to your benefit to make your project successful.

Projects may vary in terms of their scope, size, complexity, and nature. However, regardless of that, offshore or distributed teams have to be properly coordinated and managed. Agile project management framework offers several options for managing remotely developed projects.

Agile Frameworks



Scrum

Recommended for developing small to medium sized projects using a team of 7 to 12 cross-functional and multi skilled individuals. The Scrum framework is characterized by its clearly defined events, artefacts, roles, and process which have to be followed by the entire team. The error correction and retrospection activities take precedence over documentation and delegation of authority. The client is actively involved in verifying the development carried out by the team. The Scrum team delivers the business value in the project through successful product increments developed through periodic cycles known as sprints.

Extreme Programming (XP)

Extreme Programming (XP) offers a practical approach to program development and focuses primarily upon the delivery of business results. It follows an incremental, start-with-something approach towards product development, and makes use of continued testing and revision processes. XP is generally recommended for short term projects, and development teams typically follow code-test-analyse-design-integrate process. XP is known for “paired” programming i.e. two developers engaged with code development and testing simultaneously. One programmer creates the code while other tests it on the spot.

Kanban

Based upon the concept of Toyota production model, Kanban offers a pragmatic approach to development by matching the actual amount of work in progress to the development team’s capacity in delivering it. The framework provides more flexibility in terms of planning options, quicker output, a clear focus pertaining what needs to be developed, and maintaining total transparency throughout the product development cycle.

Scaled Agile Frameworks (SAFe)

Scaled Agile Framework (SAFe) is a structured and prescriptive method to help large organisations and enterprises to get started with adopting Agile. It is a popular and efficient Agile framework successfully used by many companies covering various industry verticals. It is specially recommended for large sized software based projects where teams can function interdependently.

Nexus

Nexus is an Agile framework focusing upon cross-team dependencies and team integration issues. It facilitates Agile implementation in complex and large scale projects. It functions as an exoskeleton and helps multiple Scrum teams to integrate and pursue a common goal of delivering valuable product increments through sprints. Each team delivers a certain business value to the client through each product increment cycle, and the teams achieve this by following Agile principles and process. Nexus is recommended for development teams consisting of over 100 individuals.

Agile for distributed teams

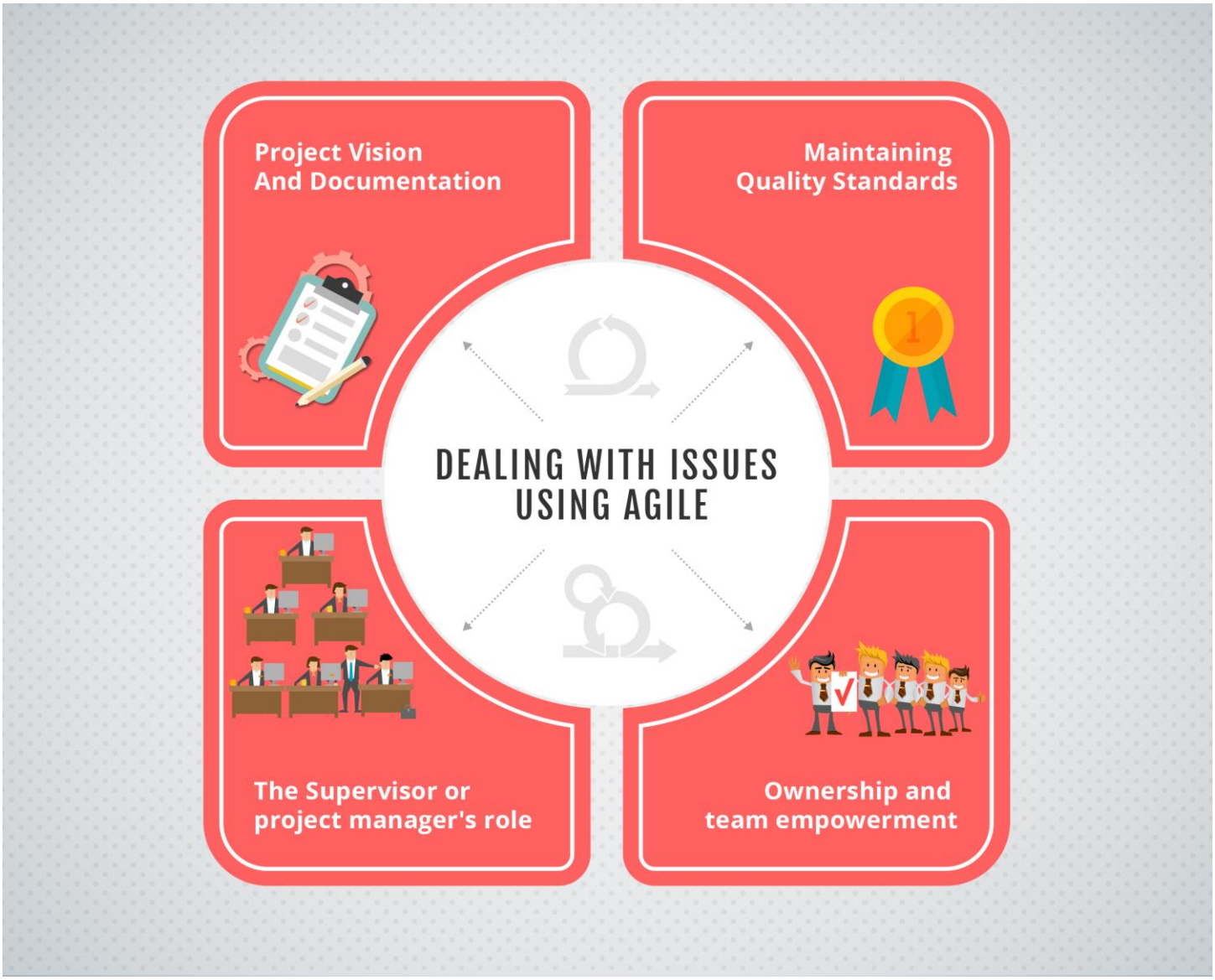
While executing your very first remote project, the most logical thing to do is to document the project vision and figure out how the team will deliver the project goals. Proper and effective communication is of paramount importance while explaining the goals and objectives to team members. It is a simple and straightforward process most of the times, but while working with distributed teams, the cultural differences and varying language proficiency levels may often create constraints and lead to miscommunication as well as confusion. This can be a common scenario in case of teams located in countries across different time zones and possess limited ability to communicate using a particular language. Individuals may find it difficult to understand and capture the exact project requirements and deliver code or functionality that does not fulfil end user requirements. Projects often fail because of these and other such technical and non-technical reasons.

Using Agile it may be possible to simplify these types of problems. Agile is not a silver bullet that can rectify all issues and problems faced during project execution. Agile is a framework, therefore It depends upon how well the team understands its principles and how effectively it implements them in

the project. However, the framework is designed such that issues can be dealt with in a more proactive and effectual manner.

Dealing with issues using Agile

Businesses opt for remote or distributed teams mainly to segregate the development activity from the main organisation body by trans-locating the team and development activity to some other location for management or financial reasons. The team is directly employed by the organisation and each member is an employee. In case of offshoring, the entire project is outsourced to a development vendor who executes the project on behalf of the client, or develops it as a part of client contract. This discussion does not try to differentiate between whether the remote team is a part of parent organisation or it belongs to an outsourcing vendor. Some common issues faced while working with both types of teams are discussed and how those issues can be properly targeted using Agile. It is worthwhile to know that Agile is not the only project management platform to develop IT or software projects. Neither does it offer a guaranteed way of dealing with issues faced while working with or employing remote teams. However, the framework is uniquely designed, and is flexible enough, to deal with such issues in a more effective manner, and more easily.



Project vision and documentation

The project vision explains the goals and project deliverables. The primary aim of the team should be to deliver work supporting the vision so meaningful business value can be delivered to the client. Often, development teams put in efforts and deliver work, but when reviewed by the client, it is discovered that the features developed don't exactly support what the client actually wants. This can be a very common scenario when teams are unclear about what the project aims to achieve and why it exists in the first place. Common reason why teams may fail to understand the vision could be language barriers (In case of distributed teams located in different countries and speaking different languages) or a lack of proper communication from the client's or management's side explaining the objectives. Agile does not emphasize upon extensive documentation. In real life scenarios elaborate or extensive documentation often remains locked away in filing cabinets or resides on shelves for future references – teams rarely bother to read them thoroughly since they can be large in size and a lot of time is spent in reading and understanding them. The attitude of most development teams (Don't mean to disrespect them in any way) is to get started with work so deadlines can be met. Teams are generally pressed for time so they don't bother, or can't afford to spend hours reading comprehensive documentation. Paperwork is greatly reduced in Agile, and if you choose to follow Agile, you need to create just enough documentation to get started with work. More importance is given to understanding client-centric requirements and delivering business value, rather than creating elaborate reports and documents. Moreover, one of the responsibilities of the product owner in Agile is to ensure that the team understands the deliverables and project vision properly before it starts to work. The PO also makes sure that the business value delivered from the sprints is useful and matches the project vision.

Maintaining quality standards

Quality and deadlines are two most important factors associated with, and affecting, the success levels of a project. Quality features fulfilling end user requirements have to be developed within the decided time so it can be properly marketed and business returns availed from it. In the IT market segment it is not just important to build quality software, but to release it in the correct manner at the correct time and at the correct place (targeted market audience i.e. the geographical boundaries within which end users are likely to buy your product. With online marketing these boundaries remain virtual but nevertheless play an important part in deciding the "target audience" when the project is planned and incepted). When outsourcing work to remote teams, the quality aspects could get compromised upon if a QA or testing process in set up as a part of development process. Fewer development teams actually bother to test the code for regression after it is developed unless it is a pre-decided activity and integrated with the development process.

The Agile manifesto states "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software." It emphasis upon "early and continuous delivery of valuable software" i.e. useful and valuable product features should be developed and delivered to the client on regular basis. Agile focuses upon the delivery of "shippable" features. Each feature should be properly tested for errors and made bug free before its development can be considered as complete and deployable. Developers and programmers often double as testers to carry out the QA part during sprint cycles. Agile fails if "workable" features are not developed. Remote teams trained in Agile have to fulfil the test conditions stated in the acceptance criteria defined for each development task created in the product backlog (ideally).

The supervisor or project manager's role

Every project needs a manager to oversee its execution and completion. It is important for the supervisor or the project manager to remain available to the team and resolve problems and issues as and when they occur. When teams are located on-premises it becomes easy to resolve technical problems since face-to-face interactions are possible and the manager is always available when you need him or her. That is not always the case with remote or distributed teams. Owing to time differences, the manager could be ending the day while the remote team would be just about to start with work. Teams may be required to wait for some time before problems are resolved, and this could delay work further. Deadlines and commitments may therefore not be met. The Scrum Master's role is very clearly defined in Agile framework. The SM often plays a servant-leader role, and mentors and facilitates the Agile process. The SM ensures that he or she is always available to the team and resolves glitches whenever the team gets stuck. In Agile, the Scrum Master is a specific role played by a person, rather than a designation or responsibilities given to a single individual. The role can be played by anyone in the team. In case of distributed teams, a responsible team member can be taught to play the Proxy Scrum Master's role and provided with quick-access channels to communicate with the actual SM or PO in case of urgent issues. The person also functions as a team representative and creates daily feedback reports which can be studied by the client, PO, and the SM as per their convenience.

Ownership and team empowerment

Traditional project management methods differentiate between senior and junior level individuals, and have a clear hierarchical structure defining authority levels and who reports to whom. Even today, most organisations still follow this traditional hierarchical model, and individuals belonging to different levels of authority remain concerned about their responsibilities and reporting status. Even though the model is organised, it takes a lot of time for issues to get resolved as the escalation process involves several individuals starting from the junior level to senior levels. Moreover, people have a tendency to "pass on" issues to senior levels personnel and let them decide what to do next. Technical staff and junior level employees may prefer not to get involved with decision making since they often become scapegoats to bureaucratic procedures. In case of distributed teams the scenario can become even worse because you don't have to deal with just bureaucratic attitudes but the language and distance factor may further make the team even less accountable for the success or failure of the project. Agile does not believe in shifting responsibilities or escalating issues. As per the model, teams are cross functional and self-managing. Each team member often takes up additional tasks other than his or her particular skillset thereby reducing the total numbers of skilled members required in the team. There are no senior subordinate levels – just three primary roles of product owner, scrum master, and the development team. Rather than assigning tasks, each team member voluntarily takes up work based upon his or experience and skills. One of the most important aspect about Agile is that the team has to "own" the project on behalf of the client. It means each person is responsible not just for the work done by him or her, but the overall contribution of all members at the team level is even more important. The entire team is accountable for the success or failure of the project – not just the product owner but each and every member of the team.

Moreover, the three roles of PO, SM, and the team are empowered in Agile to decide on their own what course of action to take to best fulfil their objectives. The development team is not required to follow orders or take permissions in deciding how a particular feature should be developed, and in what manner. It has to deliver work as decided in an event – the sprint planning meeting – held before each product incremental cycle known as a sprint starts.

References

1. Scrum.Org and ScrumInc. "The Scrum Guide." Scrum Guide. June 2013. Accessed April 29, 2016. http://www.scrumguides.org/scrum-guide.html. A sentence from the Scrum Guide is quoted in the section "Maintaining quality standards" in the chapter "Distributed Teams Challenges And Agile Advantages". Microsoft. "Agile Development - Working with Agile in a Distributed Team Environment." Agile Development - Working with Agile in a Distributed Team Environment. January 2012. Accessed April 29, 2016.

2. https://msdn.microsoft.com/en-us/magazine/hh771057.aspx. Issues pertaining to distributed teams narrated in the section "Co-location vs. Distributed Teams" found on reference website are explained within the section "Part 3 – Dealing with issues using Agile" of chapter "Distributed Teams Challenges And Agile Advantages" in the PDF.

Contact

France
Quickscrum SARL
47, Rue Klock
92110 - Clichy, France
Ph. No.: +33 - 184 88 70 42

Germany
Bharti Consulting Services
Agentur Köln
Rhöndorfer Str. 61 50939 Köln
Ph. No.: +49 - 322 21 09 42 91

India
Bharti Soft Tech Pvt Ltd
89 - Narayan Complex, Jayant Oil Mill Road
Near Vadsar Bridge
G.I.D.C., Makarpura,
Vadodara - 10
Ph. No.: +91 - 265 654 43 12