# 6.1 Planning and Execution

**Ahmad Amarneh Shruthi Chireddy Swetha Eshvar Khalid Hejazi Tejaswini Tummala**

**Published on:** Jun 17, 2020

**URL:** https://managing-remote-software-projects.pubpub.org/pub/1ml71pee

# Customers' Feedback and Expectation Management

## Introduction

As Jeff Bezos said, "If you do build a great experience, customers tell each other about that. Word of mouth is very powerful." Customer experience is a significant competitive framework because a satisfied customer not only come back but recommends and refers to others. It becomes highly vital for project managers to put into consideration users' feedback and expectations when tackling a project to ensure the best outcomes. Wordfence remote company makes security plugin for WordPress and helps in protecting the vast WordPress websites from attackers and detecting intrusions (“Remote Work at Wordfence,” n.d). Mark Maunder, the CEO, attributes multiple benefits to having a remote workforce such as more time finding a solution and multiple time zones ensuring there are people working at a given time. The CEO acknowledges the importance of having the right communication tools and maintaining constant communication with remote team members and customers (“Remote Work at Wordfence,” n.d). Wordfence utilizes Slack, Skype for communication. Mainly, tools, such as Git, along with FogBugz and GitHub allow remote members to communicate issues among themselves and interact with their customers ("Remote Work at Wordfence," n.d). Wordfence has also provided multiple email addresses for new attacks or malware, purchasing and press questions, general business inquiry, feedback, and product support, where clients can contact and interact with them. For instance, WordFence deploys FreshDesk, cloud-based ticket management software, to upgrade how WordFence interacts with clients and manages customers' feedback and queries. If the FAQ AI-feature webpage does not help customers in finding what they search for, a questionnaire forum and chat area popup to propose offline and online communication channels, respectively. With such a tool, WordFence aims to keep the team engaged proactively with its clients, and that helps in managing customers' feedbacks and expectations skillfully ("Remote Work at Wordfence," n.d).

This chapter will review practices for planning and execution on managing remote software projects with a focus on customers' feedback and expectation management. It will particularly explore agile software development, feedback and learning, and practices and challenges for remote and distributed teams at agile software development. It will also discuss how to manage user expectations on software projects, the impact of user expectations and challenges on software projects, and strategies to manage user expectations. This chapter is organized as follows. After the introduction, the chapter will feature the agile software development and will focus within that on the feedback and learning aspects. Then, the chapter will discuss the remote and distributed teams at Agile Software Development by presenting practices and challenges for remote agile teams. The next part will talk about managing user expectations on software projects with highlighting: risk and its associated impact of user expectations and challenges on software projects. Finally, and before the conclusion, this chapter will propose a list of strategies to manage user expectations remotely. **Figure-1**

illustrates a concept map that guides our valued reader to understand how this chapter is structured in line with other chapters that cover the Planning and Execution Stage.

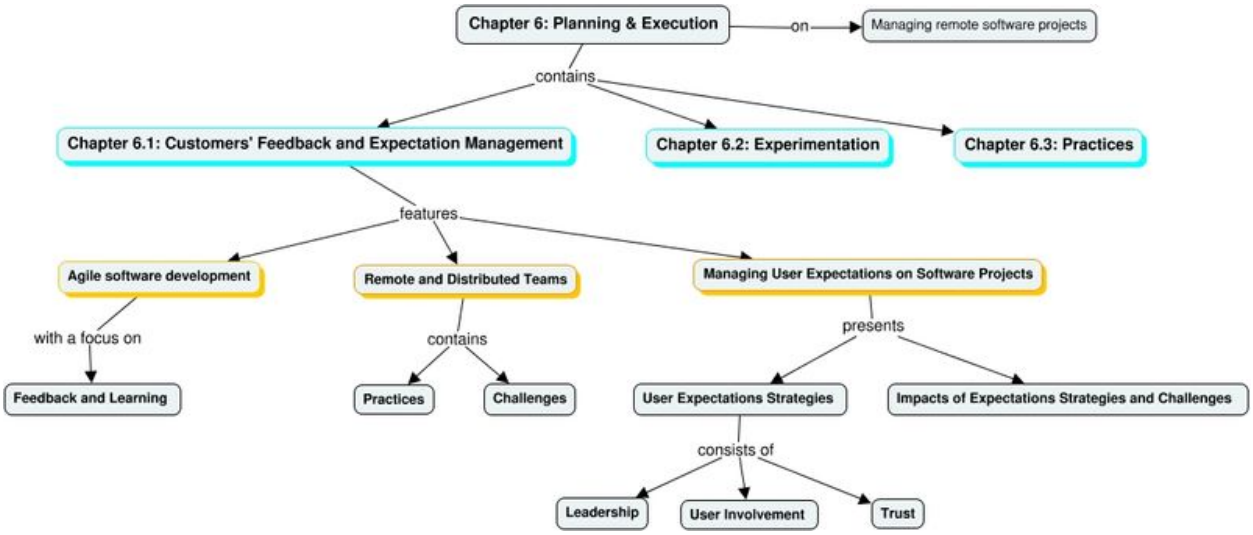


Figure-1: Concept Map of the Planning and Execution Chapters.

## Agile Software Development

A software project comprises multiple activities that result in the development of an identifiable outcome of value. Project management involves planning, monitoring, and executing project activities. Due to high failure rates and costs associated with software projects, there is a significant interest in agile software development, where the developers work in teams with the customers (the end-user). Implemented features in different development cycles are chosen and decided by the customer and the development team jointly. Managers of software development projects within the agile environment need to adopt and utilize specific management activities and principles that support them in controlling the unique processes, uncertainty, and complexity innate in agile software development projects... Dybå, Dingsøyr, & Moe (2014) indicate that the primary problems in software development are mostly managerial and not technological. Project complexity means that multiple factors such as the technological, environmental, and organizational impact a software project making it challenging to assess their actions. Managing software projects requires less reliant on past assumptions and experiences due to unexpected developments and unpredictable human behavior.

Agile project management concerns itself with managing the effects of uncertainty and complexity on a project by recognizing the need for a shorter time between planning and execution, and the necessity for creativity, learning, and feedback in completing a project (Dybå et al., 2014). It introduces a shift in management responsibilities and practices; for instance, scrum identifies three roles, including a facilitator, development team members, and the customer. Facilitators organize meetings and ensure the team handles obstacles effectively. Hyde et al. (2014) acknowledge that facilitation is complex and involves ethics and productive power. The roles of facilitators involve mediating in controversy, governing group dynamics, and

imposing deadlines. They create enabling environments and conditions necessary for purposeful and creative collaboration by ensuring members share their opinions and take part in the outcome of the project, and utilize motivation approaches and different learning styles to manage potential conflict (Hyde et al., 2014). Conversely, the project owner prioritizes and decides on what is to be developed by the self-managed team. The teams should have the ability to deal with ambiguousness presented by project tasks in terms of expertise required to complete a task, implementation of appropriate control strategy when executing a project, and coordination challenges. **Table 1** summarizes the agile software management practices.

Table-1: Agile Software Management Practices

| **Conditions supporting self-managing teams** | **Team Leadership** | **Feedback and learning** | **Principles of Agile Project Management** |
|---|---|---|---|
| Engaging and clear direction. | Shared- decision making – shared decision making and team empowerment is encouraged at all levels (strategic, tactical, and operational). | Agile Project Kick-off – describes the project team, planning of the project, and establishment of roles. | Minimum critical specification – project managers should not specify more than is critical to overall success. |
| A supportive organizational environment and context. | Escalation of commitment- decision-makers should avoid allocating resources to a failing course of action. | The Retrospective - collective learning activity after an iteration or release to improve future practice. | Autonomous teams- they are self-managing and responsible for their own processes and tasks. |
| An enabling performing unit system and structure | - | Visualizing Project Status - utilizing visual boards such as kanbans to facilitate collaboration, communication, and coordination. | Redundancy – having multi-skilled team members. |
| Sufficient resources | - | - | Feedback and learning- a vital step for executing the project effectively. |
| Expert and available coaching | - | - | - |

## *Feedback and Learning*

In agile software development, learning and feedback are essential aspects to adapt to technological changes and customer requirements. Feedback loops are found within the team as well as with customers. Don

Wells (1999) developed the XP feedback loop diagram to illustrate the different types of feedback loops in agile methods. **Figure-2** shows that feedback loops.

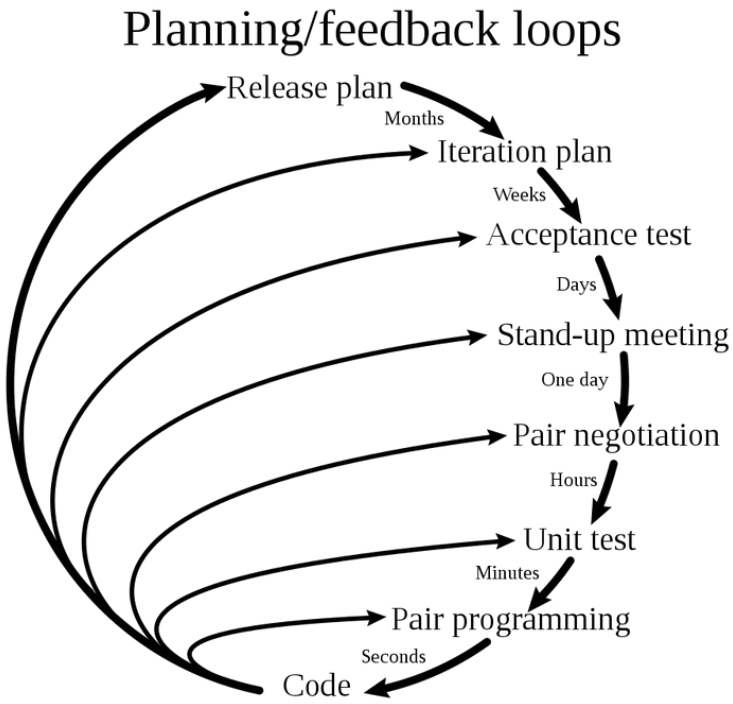


Figure-2: Planning/Feedback Loops (Wells, 1999)

Feedback offers a learning opportunity resulting in product changes, changes in the development process, and improved personal skills leading to an overall better outcome (Dybå et al., 2014). One of the primary arenas for feedback and learning is the agile project kick-off, a project management tool. During a kick-off meeting, project vision is described, the development team roles and project stakeholders are established, and planning for the project is defined. The customer or the product owner defines the goals and vision for the project and represents the project stakeholders (Dybå et al., 2014). Iteration durations are essential when planning the project, with frequent changes in technology or client requirements calling for shorter iterations, while stable environment environments allow for longer iterations. The development team makes a rough plan for multiple iterations, and a detailed plan is developed at the kick-off by allowing the customer to provide the features to be developed. **Figure-3** shows a kick-off meeting with the development team, the team facilitator, and the customer.

Figure-3: Kick-off Meeting (Dybå et al., 2014)

Different techniques are essential for enabling feedback and learning in the kick-off meeting, such as shared mental models. It involves task knowledge, skills by the team members, interaction, and knowledge of technology (Dybå et al., 2014). Planning poker technique facilitates shared mental models where individual members are given cards representing the number of hours or days one can use to complete a task and compare the estimates to understand the reasoning of the members (Dybå et al., 2014). With a clear image of team interaction, the team members have a clear work process ensuring they deliver the development cycles within the time frame given. A retrospective is a learning activity conducted after an iteration, which involves reflecting on the processes taken and what needs improvement for future practice. The set of activities includes illuminating the plans and achieved practices to understand any experience overlooked in practice (Dybå et al., 2014). The activities facilitate learning, personal and professional growth, and offer an opportunity to celebrate success. **Figure-4** shows a retrospective with developers structuring brainstorming results.


Figure-4: A Retrospective (Dybå et al., 2014)

Consequently, it is essential to visualize the status of the project using visual boards and kanbans. A physical board is easy to refer to and provides a shared mental model of task importance and time remaining to complete a project. The principles of agile project management include minimum critical specification, autonomous teams, redundancy, feedback, and learning (Dybå et al., 2014). Only the necessary system requirements should be specified, and the use of rules should be kept to a minimum. Autonomous teams involve self-managing individuals who are responsible for monitoring processes and completing tasks without supervision (Dybå et al., 2014). Teams should include individuals with overlapping knowledge to enhance the sharing of knowledge. Therefore, working with the product owner, who prioritizes and decides on what is to be developed, having a self-managing team, utilizing project kick-off, retrospective approach and agile project management principles, are some of the practices for planning and execution on managing remote software projects while focusing on customer feedback and expectation. **Figure-5** shows a Kanban board and the primary elements of the board.

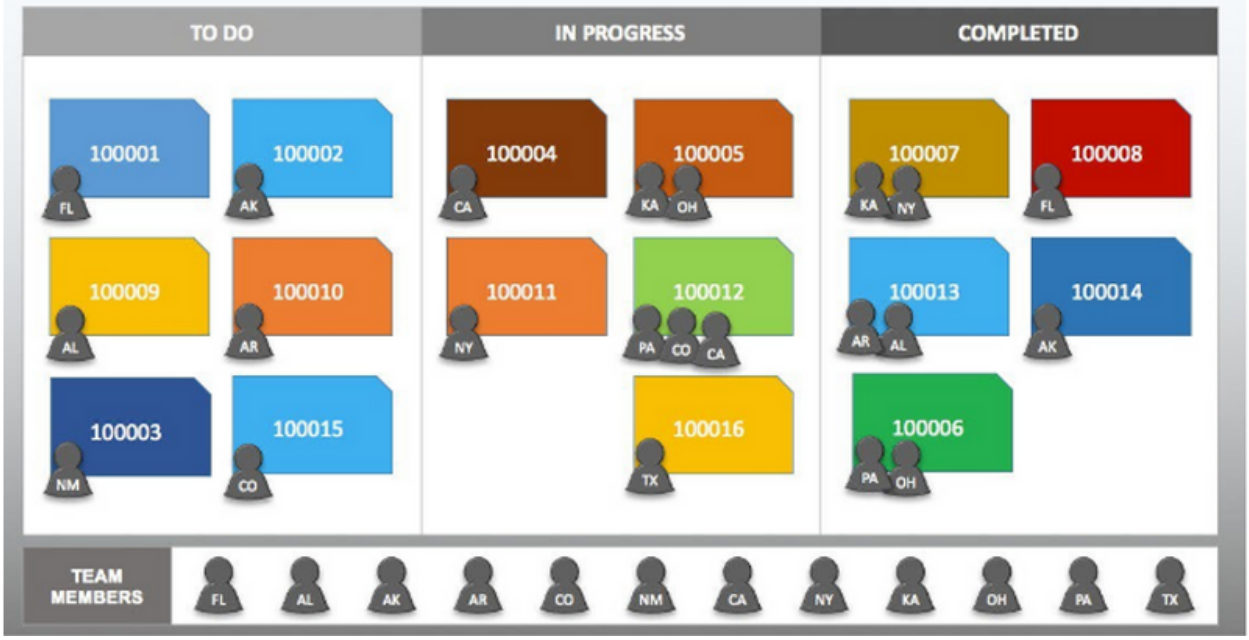


Figure-5: A Kanban Board (https://blog.taskque.com/how-can-professionals-use-kanban-boards/)

## Remote and Distributed Teams at Agile Software Development

Most organizations believe that agile methodologies work best when team members are in one location. When team members share a workplace, they easily ask and answer questions, pair on programming tasks, and solve challenges as they arise without scheduling for meetings. While it may take significant efforts and experimentation, companies can make agile methodologies work excellently with remote and distributed teams. Teams must optimize the use of technology and adopt effective communication styles to facilitate quality, productivity, and collaboration. In Chapter 5, *Working in Teams*, the authors discussed Lous et. al (2018) to present how the studied company (Debitoor) has adopted agile practices (Cadences, Ad-hoc, and Continuous) and the supported tools (technical, automation, and operational) in managing distributed teams

### *Practices for Remote Agile Teams*

**Always have multiple open channels**. It is critical to stay connected with distributed teams, which can be achieved through phone conversations that incorporate collaborative editing of Google documents, video conversations while reviewing a product collaboratively and code reviews in a GitHub pull request (Blog, 2015). Having a chat channel is vital for synchronous meetings to help in facilitating, decision-making, and connecting with other meeting participants.

**“Work with X”.** This is a vital practice as it is tempting to work alone when working remotely. Remote teams work efficiently when the members intend to "work with" each other in the teams (Blog, 2015). To "work with X" each other means that there is trust among the members, and everyone supports the other toward a common

goal. The intention to work with each other begins when members plan their sprints. Planning together for every step is critical in ensuring that members can prioritize the most important tasks and work toward completing the assigned tasks as required (Blog, 2015). Similarly, working with each other requires commitment, which is achieved by integrating scrum practice. The daily-stand-up meetings offer members an opportunity to indicate progress and state their intentions on how they intend to help the team meet its objectives effectively (Blog, 2015). The members also solve their issues together in scrums. As members are distributed, one may not want to disturb others; thus, having time after the scrum practice to resolve rising issues is critical. Another way distributed teams "work with' each other is by holding each other to standards by incorporating working agreements. An implicit or explicit agreement between development team members and the PO team is vital to hold everyone accountable for their tasks and ensure everything runs efficiently.

**Work-life blending**. Most remote team members work from home, and thus, it is essential to effectively blend work and home life (Blog, 2015). First, it is vital to signal the team members a shift in focus when one has an interruption during a meeting. Second, members can show up for daily stand-up meetings a few minutes earlier to socialize with other members and get to know each other better.

**Work lean and not cheap**. While organizations reap numerous benefits from remote teams, it is unwise to conserve all expenditures at the expense of a project. Remote employees require adequate support in terms of infrastructure that allows them to remain connected round-the-clock (Blog, 2015). Therefore, phones, laptops, keyboards, monitors, and necessary software tools should be provided to the workers from the onset.

## *Challenges for Remote Agile*

It is vital to remain aware of different things as a remote member, such as who to contact when in need, the specific person working on an activity, and meeting arrangements, which could present a challenge for remote agile teams. Maintaining connection and communication is vital to improve awareness between the members. However, as much as working remotely presents challenges, agile manifesto applies to distributed agile teams, with proper coordination and communication. Some challenges that face agile distributed teams include remote pair programming, infrastructure, meetings with many members, social interaction, working together, and sharing of knowledge and awareness. It is vital to have sufficient infrastructure for connection, but incorporating too many communication tools could confuse the team members (Sharp, Barroca, Deshpande, Gregory, & Taylor, 2016). It is vital to choose one common communication tool with the best audio devices. The members are required to ensure their working environment is not noisy, and the devices are working effectively prior to the meeting. Physical separation of the project team members causes individual, strategic, and cultural challenges (Sharp et al., 2016). However, having a shared understanding of the project's objectives and overall goal through effective coordination, adequate communication, and control mitigates the problem. Consequently, involving remote team members in a large group meeting presents multiple challenges. For instance, remote employees do not actively participate in meeting only involving audio communication as their virtual presence is easily overlooked since they are not in the same room (Sharp et al., 2016). There is a

significant emphasis on agile teams' social discipline, where members handle multiple variables and reach an agreement between them. Applying social discipline and incorporating advanced communication technology provides a good experience for remote team members in a large group meeting (Sharp et al., 2016).

## Managing User Expectations on Software Projects

One of the main risks associated with software projects is the failure to manage customer expectations. User expectation involves the assumptions held by the end-user of the system related to information systems (IS) performance. Managing customer expectation involves putting into place necessary practices to ensure the supposition held by the user for the project are aligned and realistic to the software deliverable (Petter, 2008). A project manager is required to identify and reinforce the expectations to avoid failure. Petter (2008) formed three strategies that contain the necessary practices to manage user expectations. **Figure-6** illustrates these strategies which will be explained in the coming section:

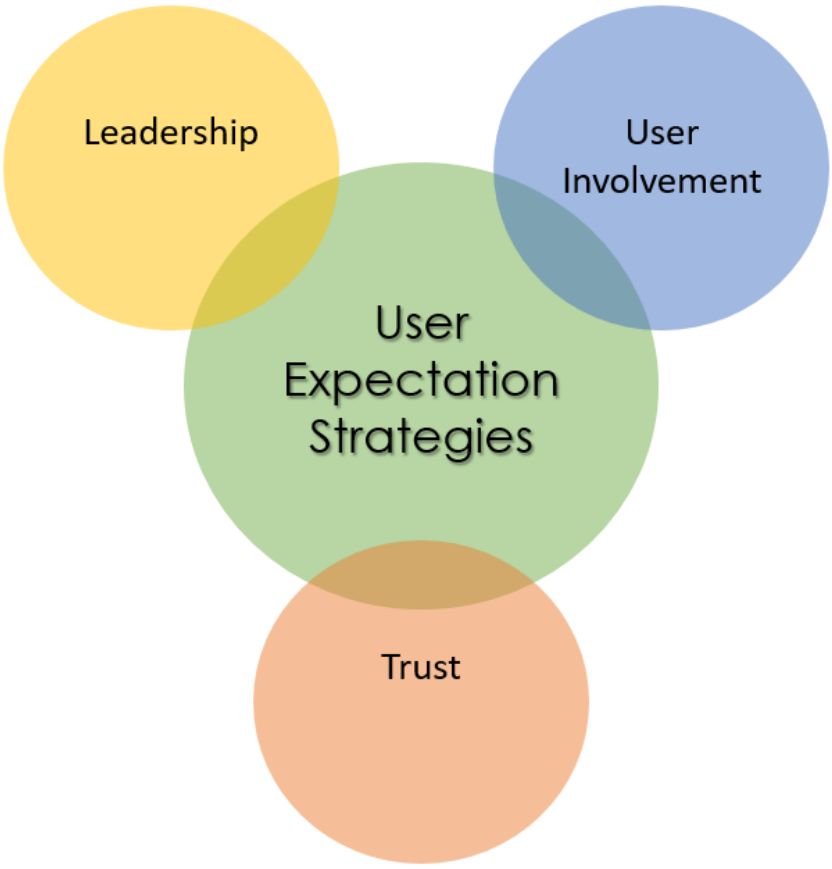


Figure-6: User Expectation Strategies

## *Impact of User Expectations and Challenges on Software Projects*

The unrealistic expectation is a significant project risk, and meeting customer expectations is a catalyst for success in software projects. Success involves project success in terms of cost, time and meeting objectives, and product success related to the outcome, end-user satisfaction, and net benefits. Ineffective management of customer expectation impacts both successes, such that if the user has a particular perception about a feature and it is not delivered as they expect, they assume the development team did not meet the functionality objectives (Petter, 2008). Incorrect expectations result in reduced levels of product success and low customer satisfaction. Expectation-confirmation theory indicates that consumers establish an expectation towards a product or a service before its use, and when these perceptions fail to match the actual performance, a wide expectation and performance gap is created (Petter, 2008). Therefore, project managers must set appropriate expectations and meet them in the software project.

When developing a project and trying to manage customer expectations, different challenges are experienced. Communication need is a significant challenge since agile development relies on informal interactions that may affect the balance in communication formality (Ramesh, Cao, Mohan, & Xu, 2006). There are numerous ongoing negotiations between the customer and the developer to determine the acceptable quality levels, which may result in unrealistic expectations if not managed effectively. Ramesh et al. (2006) described how large companies had adopted agile development in distributed content, and they organized the study around three sets of challenges, which are connected to three concepts: Communication, Control, and Trust. **Figure-7** demonstrates a concept map that summarizes and gatherers the main concepts from Ramesh et al. (2006) study.

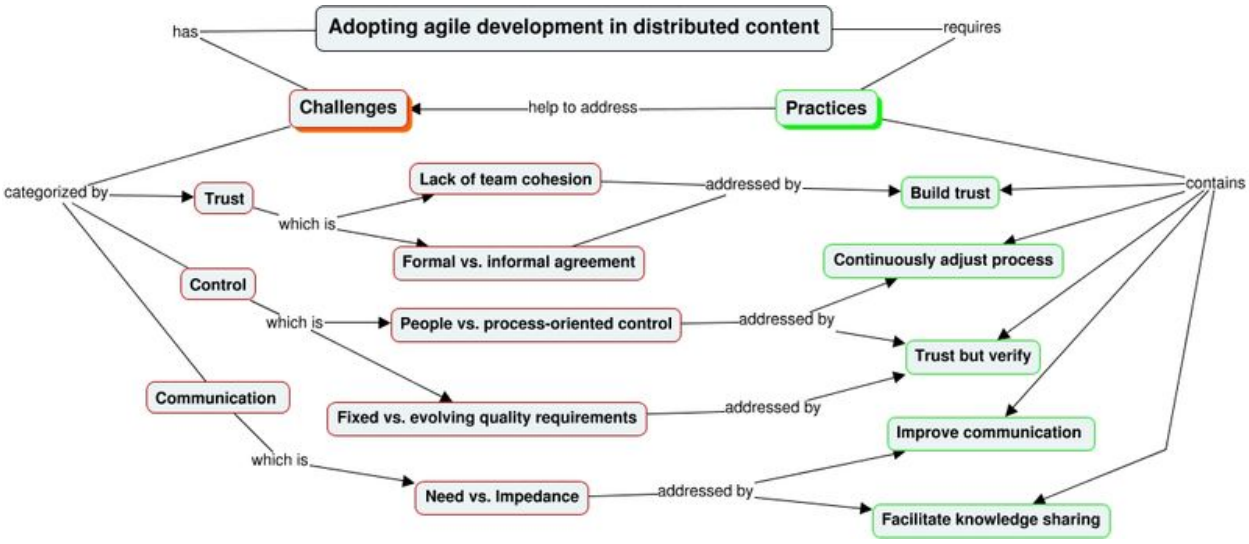


Figure-7: Concept Map of Challenges and Practices of Adopting Agile Development in Distributed Content (Ramesh et al., 2006)

## *Strategies to Manage User Expectations*

**User involvement**. Working interactively with the customer is significant to ensure the project manager grasps the needs of the user for the software. User involvement should be user-focused in that the project manager listens to their concerns and asks questions to understand more their expectations and manage them. When the project involves new processes and technology, users could be resistant to change, and managers are required to help them understand they will take them through the shift to ensure a smooth transition (Petter, 2008). The project manager is required to advise the user on different aspects such as schedule, budget, and functionality, and let the customer make any necessary choices on tradeoffs in the aspects when required. For instance, when the user requests for additional functionality close to the release date, which adds more cost and affect the quality, the project manager could suggest an alternative approach such as extending the release time. The approach would maintain the cost but add more time to complete the project, and the user should be allowed to make the tradeoff and communicate a decision.

When working with large teams, breaking the groups into smaller ones enables the members to air their needs and concerns and report findings to the larger team. It is vital to give credit to the end-users to ensure they feel like part of the team and build positive momentum from the development phase and push towards completion of the project (Petter, 2008). Training and establishing help desks and support functions are critical to providing continued comfort to the users to ensure they feel involved. Consequently, it is essential to maintain communication throughout the process to keep the customers informed and keep their expectations in check (Petter, 2008). Constant communication involves correct documentation of users' requirements, reporting after completing the design, and presenting the design to ensure it is in line with their expectations, and again after completing the development, before system testing and every other important step throughout the project.

To ensure effective user involvement, it is critical also to improve communication quality among the team members. In remote software projects, members in most cases are located in different time zones, which reduces the chances of synchronized work hours (Ramesh et al., 2006). Separation by different zones creates a communication challenge that could result in miscommunication. Implementing informal communication approaches through formal platforms to enhance coordination and balanced coordination. Constant communication is also crucial to ensure team members identify issues, invite opinions and critiques, and track the status of the project (Ramesh et al., 2006). Ensuring team members are well-coordinated and aware of the project status is essential in managing customer expectations as the project manager can report developments in the project to the users effectively. **Figure-8** illustrates the five processes of user involvement:

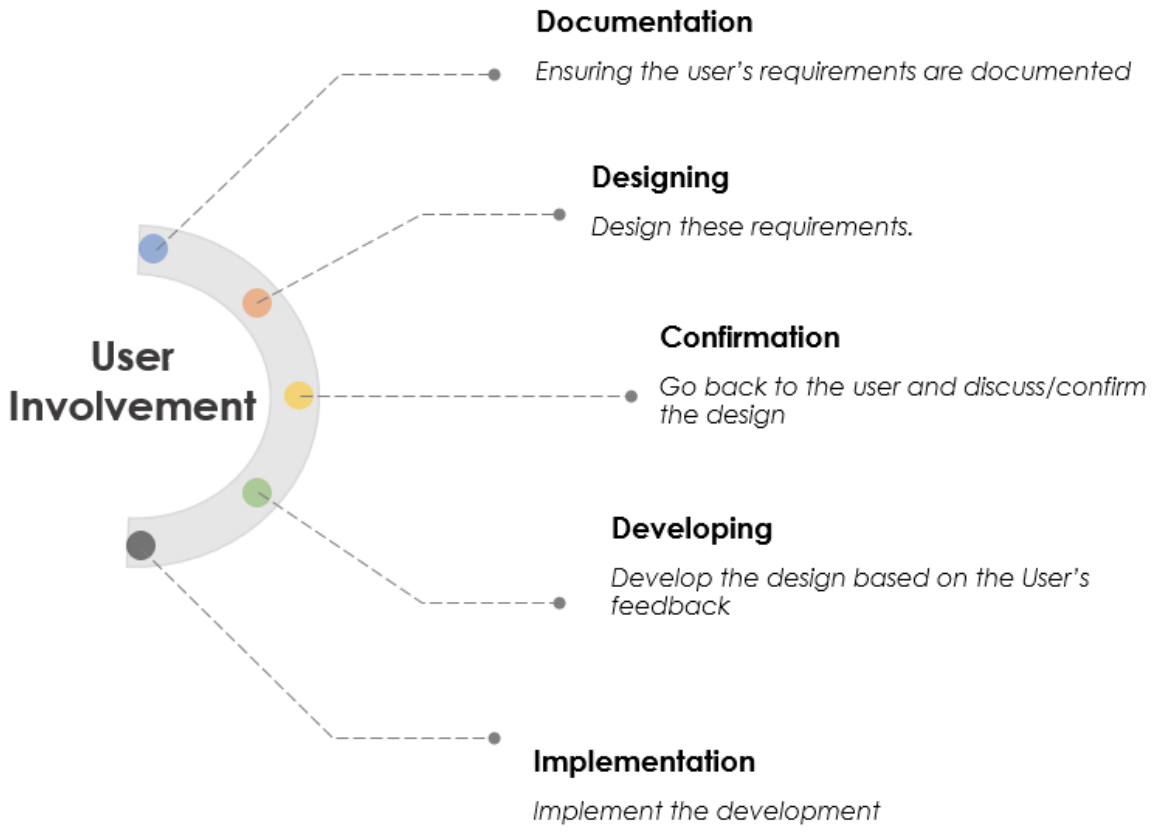


Figure-8: User Involvement Processes

**Leadership**. To ensure efficient management of user expectations during a software project, it is vital to incorporate two leadership types, including project manager for the team and a project champion for the users (Petter, 2008). The project champion is mandated with managing expectations by helping users understand the project vision and educating them on its benefits and values and their input throughout the project. Having a person from the user group who understands software values is vital to educate other users on the vision and system requirements and priorities (Petter, 2008). Project champion helps users remain at ease during the transition to the new program and encourages them to give maximum input. Petter (2008) attributes characteristics such as strong leadership qualities, project-oriented, and with clear goals for the project to the project champion. By understanding the value and purpose of the system, it becomes easy to manage user expectations.

A software project manager with strong leadership qualities is critical to managing customer expectations. A project manager is responsible for ensuring teams work as required as well as lead users to the right path by defining realistic expectations. Being empathetic as a project manager is essential to demand credibility team when making tough decisions about the project. Proper documentation and process should be followed to avoid confusion about the overall process and what is supposed to be delivered (Petter, 2008). Documenting requirements at different levels of the project by utilizing user stories and short use cases instead of detailed

use case specifications are useful for documenting and clarifying requirements. A project manager should focus on delivering value to the users by building an environment where developers and customer representatives are acclimatized to tools, processes, and applications, ensuring every individual understand the functionality the project aims to achieve in the early iterations easing the learning curve (Ramesh et al., 2006). A strong project leader should be willing to tell the customer the truth regarding the project, even when it is not pleasing to reduce unrealistic expectations. Consequently, a leader should trust their judgment rather than following advice or guidelines, which is evidently incorrect.

A project manager should realize the importance of knowledge sharing within the teams to facilitate learning. Teams should incorporate people from multiple sectors, exhibiting different expertise, skills, or even language (Hyde et al., 2014). This can be used as a buy-in technique and encourage the transfer of knowledge enriching the project. Hyde et al. (2014) equate the role of a project manager to a facilitator, who ensures creates an enabling environment for the team members to collaborate creatively. The facilitator ensures that members share ideas and engage throughout the process to achieve the intended outcome. Therefore effective leadership in terms of a project champion and project leader is vital in managing user expectations and developing a quality software program.

**Trust**. Maintaining trust is a critical element in managing customer expectations. Software project managers should communicate with users when faced with undesirable situations during the process (Petter, 2008). For instance, when users select technology before providing specific requirements, a project leader can force a solution due to mismatched requirements and technology. In such a case, the leader should share the challenge with the users to establish a solution together instead of forcing the solution to fit the requirement without communicating the problem to the customers. Hearing about the problems in the project from the software project leader rather than from another person enhances trust between the parties. Software projects present a major impact on the users' work, making them skeptical about the projects; thus, effective communication regarding good and bad aspects is crucial (Petter, 2008). Being specific is essential; for example, project managers should inform the users that they will receive the design in one week or by a specific date instead of stating soon to establish trust. When software project leaders neglect to inform or communicate with the customer of existing problems, projects tend to go awry, and trust is weakened between the users and project manager.

Organizing frequent visits by customers with the development team is essential in building trust. An organization can achieve this by having analysts and developers in customers' sites in early development cycles and when implementing important changes in the project. Sponsors visits are vital where senior managers for the sponsoring corporations visit development teams to establish ground rules, clarify requirements, and finalize contractual arrangements at the beginning of the project and to evaluate progress in different milestones (Ramesh et al., 2006). The visits help establish trust between the project manager and the users and among the senior team members. Building trust among the team members is also crucial, and the project

manager should ensure teams consist of members with prior working relationships to facilitate a cohesive team culture (Ramesh et al., 2006). As much as trust is vital, it should be verified by ensuring the teams follow acceptable practices to facilitate the required outcomes. Consequently, it is essential to supplement any informal communication with users with documentation (Ramesh et al., 2006). A project manager should not fake confidence while struggling with a project as it results in reworking, added cost, and wasted time, which breaks trust with the users. Therefore trust is an essential component of managing user expectations.

Building trust among teams and with users requires a healthy culture. Berkun (2013) indicates that no technique or approach is good enough to make workers trust each other and the managers if there is an evident reason not to. Cultivating and implementing an honest culture is essential where employees engage in knowledge sharing and are accorded trust by their seniors to complete assigned tasks in their own method without supervision. Project managers should make decisions based on what is best for the project by being guided by elements such as transparency, meritocracy, and longevity. Transparency involves having public decisions, discussions, and internal debates ensuring people air their opinions, concerns, and critiques. When people are willing to say something in front of others, they have conviction about it, and it becomes helpful. Having few designations and job titles is a common practice for self-managing teams (Dybå et al., 2014). This follows that those members who make more effort and considerable contribution and fix things automatically receive more respect than those who merely complain. These elements ensure there is trust among team members, which results in ease in building and maintain trust between the project manager and the users, which leads to the primary goal of maintaining customer expectation through trust. **Figure-9** summarizes the main aspects of managing user expectations.

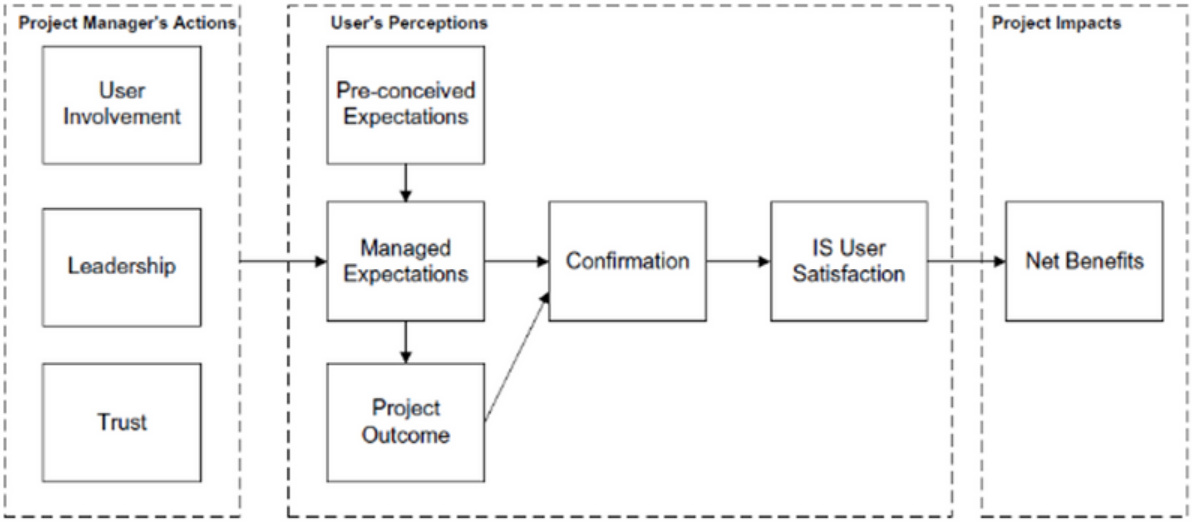


Figure-9: The main aspects of managing user expectations(Petter, 2008)

For each project manager's actions to manage user expectations (User involvement, Leadership, and Trust), there are successful tactics & failed tactics. **Figure-10** shows these tactics.


Figure-10: Tactics to follow and avoid in managing user expectations

## Conclusion

Establishing and maintain customer satisfaction is essential as clients are powerful assets. Practices for planning and execution on managing remote software projects focusing on customers' feedback and expectation management are vital. During planning, project leaders can incorporate practices such as working with the product owner, who prioritizes and decides on what is to be developed, having a self-managing team, utilizing project kick-off, retrospective approach, and agile project management principles. Ineffective management of customer expectation impacts project success and lead to reduced customer satisfaction. Different strategies to manage user expectations include user involvement, leadership, and building trust. Incorporating the components ensures effective customer feedback and expectation management.

## Assertions

1. For users' involvement, we need to keep the users in the loop as a part of the system by working with them interactively, listing to them, understanding their concerns, and updating them frequently (Petter, 2008).
2. For managing user expectation, two project management style should be incorporated: project manager for the team who obtains buy-in from key stakeholders, and project champion for the user who shares the project vision; all that lead to ensure efficient management of user expectations during a software project (Petter, 2008).
3. To identify and reinforce customers' expectations, proper alignment of agile practices is required to be ensured; and that helps in managing customer expectations (Dyba, Dingsoyr & Moe, 2014).
4. For feedback and learning processes in agile software development, the kick-off tools are used where the development team creates a rough plan. Then a detailed plan is developed to ensure proper iteration duration (Dyba, Dingsoyr & Moe, 2014).
5. The trust built between the teams helped in reducing the formality of agreements and thus enabled the development teams to rapidly adapt to the project's changing needs (Ramesh, Cao, Mohan, & Xu, 2006).

6. The practices promoting knowledge sharing gain a shared understanding of the business domain and the functionalities required (Ramesh, Cao, Mohan, & Xu, 2006).
7. For better communication orchestrating among the agile team and the users, the role of facilitators is required to exist who involves mediating in controversy, governing group dynamics, and imposing a deadline, and that ensures creative collaboration among the team and the customers (Hyde et al., 2014)
8. For knowledge sharing, cultivating and implementing an honest culture is essential, Incorporating people from multiple sectors, and exhibiting different expertise and skills are required in order to get a high quality of work, and that meet customers' expectation (Dyba, Dingsoyr & Moe, 2014)

## References

Berkun, S. (2013). *The year without Pants: WordPress.com and the future of work*, Jossey-Bass

Blog, A.A (2015). Can you be remotely agile?. *Agile Alliance*.

Dybå, T., Dingsøyr, T., & Moe, N. B. (2014). *Agile project management. In Software project management in a changing world*. Springer, Berlin, Heidelberg.

Hyde, A., Baker, R., Berry, D., Brokering, M., Dieter, M., French, A., & Rühling, B. (2014). On book sprints.

Petter, S. (2008). Managing user expectations on software projects: Lessons from the trenches. *International Journal of Project Management*, 26(7), 700-712.

Ramesh, B., Cao, L., Mohan, K., & Xu, P. (2006). Can distributed software development be agile?. *Communications of the ACM*, 49(10), 41-46.

Remote Work at Wordfence. (n.d). Retrieved from https://remote.co/company/wordfence/

Sharp, H., Barroca, L., Deshpande, A., Gregory, P., & Taylor, K. (2016). Remote working in an agile team. *TheAgileResearchNetwork.*

Wells, D. (1999). Retrieved from http://www.extremeprogramming.org/introduction.html