

Thriving with FR-Agile (Fitted-Requisite-Agile)^{1, 2}

Chaitali Chheda

Introduction

Agile is one of the most popular methodologies throughout the technology industry in the current times. It has gained prominence in the last decade mainly due to faster time-to-market, increase in the demand of nimble development of products and its promise of better utilization of people on a project team.

Scrum is one of the agile frameworks that is most popularly used for managing complex work in the software industry. The Agile Manifesto encompasses twelve key principles that are its defining principles. While most of the newer businesses build their corporate culture to adapt to Agile, it is impossible for the older corporations to completely re-structure their corporations to fully transform.

This paper will supply its audience with an insight into executing strategic, innovative projects when Agile and Waterfall do not work. The author provides an alternate methodology that combines the features of Waterfall and Agile, called FR-Agile (Fitted Requisite Agile). The suggested audience is the Project Management community. However, anyone connected to Information Technology (IT), particularly those using Agile in their current roles will benefit from this information.

We will presume that the audience is familiar with Agile and Waterfall (Phase-Gate approach).

The Waterfall or Phase-Gate Approach

The Phase-Gate approach has been used since the 1940s in the manufacturing industry. The Waterfall approach is its slight variant that was originally defined by Winston W. Royce in the 1970s. It divides work into stages or phases like Initiation, Planning, Design, Execution, Testing, and Closure. Deliverables from every stage are reviewed and approved prior to the start of the next stage. It was adopted in technology computing mainly to complement the way we worked with

¹ *Editor's note: Second Editions are previously published papers that have continued relevance in today's project management world, or which were originally published in conference proceedings or in a language other than English. Original publication acknowledged; authors retain copyright. This paper was originally presented at the [14th UT Dallas PM Symposium](#) in May 2022. It is republished here with the permission of the author and conference organizers.*

² How to cite this paper: Chheda, C. D. (2022). Thriving with FR-Agile (Fitted-Requisite-Agile); presented at the 14th University of Texas at Dallas Project Management Symposium in Richardson, TX, USA in May 2022; republished in the *PM World Journal*, Vol. XI, Issue VIII, August.

computers. Before the evolution of PC computing, writing and testing code was done through punch cards. The most efficient way to save costs was to ensure completion of most of the planning and scoping activities upfront. This was believed to result in lesser rework, lesser usage of the punch cards, thereby saving capital and labor.

A similar mindset carried forward up until the 1980s when quality of software became a key denominator to distinguish the large number of global software exporting establishments.

With the Quality Management System (QMS) principles gaining popularity, ISO first published its ISO 9001 standard in 1987 and several revisions until 2015. CMMI (Capability Maturity Model Integration) was introduced around the same time in 1987, with focus to provide guidance for developing and improving processes to meet the business goals of an organization.

At the core of these quality and performance principles, was the fundamental thought that active leadership engagement was required at all levels of an organization's development and change processes. It led to an increased mandate of leadership approvals from all stakeholders at every stage, even if they were not actively involved in the day-to-day activities. As the software industry adopted these standards, it solidified their adoption of the Waterfall approach that closely aligned with these principles by facilitating a discrete approach of managing projects through a system of interconnected stages (Initiation, Planning, Design, Execution, Testing, Closure) with gates at the end and start of the subsequent stage. Approvals were enforced at every gate to ensure that the project could progress to its next stage without having to redo any activity from the previous. In the event a phase needed rework due to a change in the customer's priorities, the project would require a considerable amount of change requests with approvals, resulting in re-baselining. The team would then work to meet the new baseline. The result was not just an addition to the planned schedule and budget, it also imposed excessive restrictions on innovation and flexibility. In a bigger picture, more workforce was leaning towards '*a process-compliant mindset*', leaving behind any scope for creativity.

The Agile Way

Agile is an iterative methodology for the development of projects and products to help teams deliver value to customers by faster time to market with lesser rework. It fosters continued team collaboration, colocation and self-organization. It uses shorter cycles of incremental delivery instead of one major release of the complete product. In a traditional project, processes are given more importance and relied upon. Agile uses an empirical approach whereby progress is achieved after observation and team consensus, instead of adhering to the defined processes.

The Agile Manifesto is the pioneering reference for Agile implementations. It seems to assume that the project in question is a complex software development project, where 'functioning software' is the primary measure of progress, the sponsoring organization will have the necessary flow of funds to deliver incremental improvements to the development effort and therefore the project will focus on the finished software product without the stringent time-boxing limitations. With Agile, individuals and interactions take precedence over processes and tools. Finished

product and customer satisfaction take a higher priority over documented requirements. Collaboration with all stakeholders is necessary and comprehensive paperwork is avoided. Agile has proven to be more adaptable to the ever-changing needs of businesses, even in non-technological industries like construction, energy and food. With the Internet and Mobile industries dominating the world, there has been a definite shift in the technological culture where most of the current operating models focus on faster time to market with the least amount of investment. It has been challenging for most of the older corporations with established channels of operation to adhere to faster paces to compete in an industry.

Ideal use cases for Agile are initiatives involving higher degrees of complexity that require a unique, innovative solution to be delivered over time or a product development initiative that usually needs faster time to market for the prototype but is improved upon gradually. Startups mostly favor the Agile way of working as it enables innovation, with lower capital investment, lower cost of labor and faster turnaround times.

Introducing FR-Agile

Agile has always been a super-effective way to work, and the number of teams and organizations leveraging it has grown significantly during the pandemic. But Agile is not the solution of all problems. Adopting Agile is not going to guarantee a successful transformation of mindset, faster time to market and an increase in revenue. A culture of ownership and open communication, support from leadership, investment into continuous training, consistence in all sectors across an organization – are some of the most important factors that lead to success with Agile delivery. There are other common scenarios where Agile may pose challenges.

Major corporations with thousands in the workforce, that adopted the standard Project Management practices and established in-house departments called the PMO (Project Management Office), still employ a process-oriented approach that will not be abandoned for a long time. The PMO processes and procedures are typically aligned to the Waterfall way of working and can be extremely bureaucratic and process heavy. If we apply the Agile fundamentals to PMOs as-is, the result can be a very chaotic team leading to very less productivity. Finding the right balance between the traditional processes and Agile is the key to produce a highly productive team, while aligning the PMO to adopt an Agile mindset. The adherence to the established processes is mandatory either for reasons related to compliance in a highly regulated industry, or to satisfy requirements from customers. Staying profitable and ahead of one's competition is a common denominator that every business needs to comply with and cannot be overlooked.

The current global software industry prefers to work with Agile over any other methodology. In the alternate line of thought - non-technical industries (like manufacturing of clothes, small appliances, shoes, etc.) involving minor scales of computing technology that require the division of their products into smaller modules during development, may be sourcing the individual modules from various vendors and assembling the unit later at the parent facility. The individual modules are generally created independently without involving the entire team, thus diluting the

basic premises of colocation, collaboration and empiricism – which form the pillars of Agile. Global outsourcing adds a further layer of complexity involving extensive protocols for exporting the finished goods, where the team that assembles the final product has no knowledge of the manufacturers of the various parts or its customers.

In both the above scenarios, organizations may consider adopting a hybrid approach that combines both Agile and Waterfall - what we call – ‘**FRAgile - Fitted Requisite Agile**’ - a flavor of Agile development that is relevant to them. FR-Agile is generally used throughout an organization or in tactical initiatives that need a faster turnaround time. For the sake of this paper, we will limit our discussion to the former scenario pertaining to the Information Technology industry.

Integrating Agile with Waterfall

Integrating Agile into an existing Waterfall methodology is a challenging process. Key factors include – transformation of processes, support from leadership, imparting the necessary training, establishing new processes and adhering to them, and most importantly – attuning the mindset of the entire workforce to this change. At a deeper level, we may face specific challenges like changing the planning and release cycles, allowing people the time to train on the new processes and making dedicated coaches available to train strategic staff, all from its inception until maturity. In addition to the extensive budget, this requires a continued commitment from all the leaders and dedication from the stakeholder community, including customers.

Full integration of Agile into Waterfall is not always achievable due to the internal as well as external risks that businesses face. It is not likely to provide any additional value add in cases where there are no complex problems in the form of a product or a project. In other words, repetitive activities like generating payroll, installing patches in existing servers, routine and preventive maintenance activities cannot gain much from transforming into Agile. But every business has departments that explore new markets, generate prototypes and launch pilot products into the market. It is these sections in a corporation that are the most positively impacted in an Agile setup when developing and designing prototypes. Organizations that do not find it feasible to transform to Agile completely, can look at an alternate path of transforming their select R&D departments or new product developments to Agile, while still adhering to the established processes at the corporate level.

Any gradual transformation into Agile is beneficial in the sense that it gives teams enough time to identify the processes that no longer work to create newer ways of collaboration and empiricism. This can be especially significant when applied to the R&D or new product development labs because they have more leeway to execute in sprints and determine what will work in the end. These teams may or may not use all the tenets of Agile. Time-sensitive tactical projects can also gain from this approach. They can easily spin off dedicated teams for the specific objective. The Agile ceremonies are usually tailored to meet the nature of the problem, to increase productivity in a shorter duration.

Proposing FR-Agile

A major Financial Services' customer was in the process of transforming its operations from the existing on-premise (legacy datacenters) to the public cloud. In this traditional, process-oriented organization it was anticipated as such a major technological leap, that the Chief Technology Office (CTO) sector was chosen for its planning and execution. The idea was to establish a Public Cloud Platform as MVP (Minimally Viable Product) in six months and migrate four, non-critical, pilot applications onto it. The remaining applications would then be slated for adoption, eventually transitioning all applications to be hosted from the native data centers to the public cloud. The main idea behind the planned transformation was to decrease this outdated, bureaucratic Waterfall system that hindered releasing products at the required pace. The complete lifecycle of a new product that took more than eight months from inception to go-live, with the current procedures and compliance standards, would be immensely shortened. *The future vision of the leadership was to employ technology to transform the culture of the organization to that of a modern startup with increased employee engagement, innovation and empiricism.*

A team comprising of the best in the company was formed. The forty or so team members knew that they were trying to achieve what seemed like an impossible objective while working against the existing cumbersome processes and approvals at every step. They were motivated by the purpose and that proved to be an important starting point. The entire team struggled for the first month or so and finally decided that in order to succeed in the short duration, some things will have to change. A list of the proposed changes was created and presented to the leadership.

Following is the initial, proposed list:

1. The entire program was to be executed as non-traditional initiative, owing to the complexity and unknown factors. Another significant reason was to steer away from Change Requests and the burden of associated documentation. The team was still considering Agile as the most suitable option.
2. All interfacing sectors within the organization, including the Public Cloud Vendor organization would need to adapt to the processes laid down by this team, for the sake of being consistent. This was crucial as the vendor team was not yet defined. We had only initiated discussions about a contract.
3. All members of this core cloud transformation team would be dedicated solely to this effort. All members were to be relieved of their usual (Business as Usual) responsibilities until after the completion of the program.
4. The core cloud transformation team would be allowed to bypass all the usual channels of approvals.

5. A dedicated space with fifty cubicles would have to be allocated to the team to be collocated, to enhance collaboration.
6. A war room with video conferencing service was to be allocated to the team for the entire duration of the program. This would serve as the central location for the team to meet as frequently as needed, without having to go through the official channel of having to reserve meeting rooms.
7. Weekly Steering Committee meeting was scheduled between the Program Manager / Product Owner and all the CTO Managing Directors and above. The purpose of this meeting was to present any major roadblocks and resolve issues as they arose by seeking assistance. Any communication from the leadership was also shared with the executing team, so there was a constant channel of interaction between the core team and the Senior Leadership Team (SLT).
8. The core transformation team was broken down into multiple work-streams. Each work-stream had a dedicated purpose to achieve like Network Connectivity, Infrastructure as Code, CI/CD (Continuous Integration / Continuous Deployment), Application Integration, to name a few. Individual work-streams met as frequently as required in a day. A single point of contact was appointed as the lead to triage the tasks in every workstream.
9. Leads from all the workstreams could meet with the Program Manager at the beginning and end of each day. This was optional; only for situations when an open issue needed to be addressed as an escalation.
10. A working list of risks and open issues was created in MS SharePoint, with dates for resolution across every item. This list was an interactive list that was accessible to every member of the team, as well as to the senior leadership. It was reviewed daily to track progress across all sectors of the organization that had any role to play. Clear owners were assigned across every item.
11. The Product Owner / Program Manager served as the single point of contact for all external communication and as well as outside of the core cloud transformation area, with the SLT and Vendor management.

The Forming, Storming and Norming Phases

Psychologist Bruce Tuckman came up with the phrase “forming, storming, norming and performing” in his 1965 paper titled – “Developmental Sequence in Small Groups”. We were coming together as a brand-new team for a crucial objective. The ‘forming’ was the start of drifting away from the Waterfall method. In the first few days, the team spent time to achieve clarity of overall objective, creation of a new platform and what it meant at the deeper level. All the pre-

requisites were discussed and as we progressed further, we started identifying the sectors outside of the CTO that would have to contribute to our work as dependencies. Key contributors who had planned time off from work in the next three months discussed their options and provided backup staff. By now, the purpose was understood, roles and responsibilities outlined, general team guidelines were established, and commitments sought, thus concluding the ‘Forming’ stage.

The next stage was the Storming Stage, where we dived deep into the numerous specifics. While working non-traditional was a novel idea to most of the team, it was of extreme pertinence in the current scenario. In the absence of any proper structure, we had to discern every minute detail for the success of the initiative.

Secondly, the Public Cloud Platform was being developed for the first time in the history of this company, we could not use any existing architectural designs. Any new designs that failed during integration with the existing applications, had to be revisited all over again and this was to be done in short sprints, to stay on track with the release timelines. With the cloud technology emerging at a rapid pace, it was essential to quickly adapt to the latest services, configure the services to function with our applications, test and integrate. All this would need to happen within hours or days as opposed to a phase. Waterfall could not work in this case. Agile was the consensus of the team.

The idea of trying to adopt Agile came with its own set of challenges. About 70% of our team did not have any real experience with Agile, the User Story creation, Epic-writing and Scrum events. It was a favorable decision at the outset. A sizeable portion of the core staff believed that a ‘truly Agile development’ is an ongoing effort without any end-date that would provide the opportunity to work without ‘a schedule’. Agile is a suggested framework and not a compliance standard as in the Financial Services’ industry, so it could follow the sprints without any solid outcomes and still be compliant. We even deliberated for hours on Kanban vs Scrum, leaning towards the former as Kanban was a simpler option. The team did not agree to Kanban because that using it would mean a lot of boards to keep up with. For every workstream, came every sub-workstream and its own tasks that we had to account for. We estimated an excess of fifty boards to track.

Just as the executing team had presented its challenges, the SLT added its own by asking for a Program Plan that would guarantee that the proposed implementation date was being met. This was particularly stretching as the we did not have a clear line of sight on the design approach. The design was volatile and so was our plan.

The other major challenge was in establishing that the usual channels of approvals be bypassed for us. The Senior Leadership Team was completely divided over this because it posed a significant risk to the organization’s Cyber Security, our Customers and the Risk Management and Compliance initiatives in the company. Being part of the financial services’ sector, this was non-negotiable.

FR-Agile to the Rescue

The obstacles we were facing needed acknowledgement and a tactical resolution. As a team we debated what would and would not function. We deliberated over the trade-offs that we thought would propel us ahead with some flexibility. We were at a stage where the team was comfortable to voice concerns but was open to accept the impediments and find solutions as a cohesive unit. We sought structured constructive feedback from the SLT across all the sectors. In some exchanges we were asked to provide a working plan, including risks and issues and the corresponding mitigation steps. We could only provide this data for the known risks.

Following is a summary of the decisions we arrived at, as the tactical resolution for this effort.

1. We would not employ Sprint Planning or Retrospective. Most of the team would need extensive training in the Agile fundamentals. We did not have any cycles for training. We would set half a day as a ‘Demonstration Day’ at the end of every sprint, which would coincide with the end of a release cycle for the rest of the organization.
2. We would follow sprints (shorter cycles of incremental development), but they had to be aligned to the corporate release timeframes. Instead of the proposed bi-weekly sprints, we had to follow the monthly sprints to coincide with the monthly release cycles.
3. The team could not be left to make all the decisions, as most of the decisions revolved around the resulting Enterprise Architecture, and we had limited cycles before implementation. The SLT had to approve before we could proceed.
4. After much discussion and deliberation, core staff was not able to create any meaningful User Stories or EPICs. Objectives were broken down into milestones and further by tasks, based on team consensus.
5. Velocity was not calculated by the tasks, but by the milestones as in a traditional Waterfall scenario.
6. MS PowerPoint presentations depicting the RAG (Red, Amber, Green) status continued to be an established medium for updates to the SLT.
7. Daily stand-up meetings continued in every work-stream. This aided the teams to be on track and bring out any immediate issues that needed to be added on to the program tracker.

Vendors and FR-Agile

Our inability to empower the vendor staff to make decisions as in true Agile, was one of the challenges. The other challenge was around billing and budgeting.

Vendor organization was educated in our organization's FR-Agile principles. It took a few cycles to assimilate their staff with our employees and establish expectations related to FR-Agile. Every workstream was assigned a vendor representative lead and an employee lead. This promoted a somewhat better flow of then latest technical data, resulting in lesser rework of design. There were several instances when the vendor staff was required to look at the legacy data to provide the appropriate solution, but we could not share the sensitive data. At times, the proprietary data was scrubbed and in alternative situations we sought explicit approvals from our compliance department.

Businesses involving technology for large banks and Financial Services generally look for a fixed pricing structure with for every software development initiative. It provides budget assurance with financial accountability. With a fluid initiative like ours, where the design was changing to adapt to the legacy applications, it was nearly impossible to stay with fixed scope and pricing. We did not have any specifications defined upfront, our application service requirements were changing from free services to the paid ones (for better post implementation support) and the timelines were not exactly helping the situation. The constant release of newer services from the vendor was also a factor affecting our decisions, because at the core of our platform development, we wanted to use the latest technology without having to make frequent changes later.

The T&M (Time and Material) option was better suited to this effort, as it accommodated the continuous cycles of implementation and testing. It would resolve the challenges of changing scope but present another complication on budget and cost control. As the management was not agreeable to a complete T&M model, a hybrid option was the best way to go. We adopted the T&M model with a Cap on the total allowable budget.

A list of the key points that aid in the engaging and working with the vendors, when using FR-Agile are summarized below:

1. Provide a fair opportunity to the vendor staff to understand the culture and norms of the client organization. It must also be made aware of all the known risks and issues.
2. Establish clear objectives of the deliverables and timelines, including what constitutes 'done'. In most cases we see clients diving deep into the technical deliverables during vendor orientation. At this stage, it is likely that you will not have full design requirements. Keeping the discussion limited to the objectives saves you from future disruptions involving budget and personnel, should the technical approach require a considerable change.
3. Establish metrics and governance that relate to an outcome. The vendor team must clearly understand the expected outcomes and provide a daily, weekly or a bi-weekly update on the progress.

For example – In one of the weekly status reports from the vendor, it listed the release of an addendum to an existing service as an accomplishment. For us, the client organization, this was not considered as ‘a measurable outcome’. The successful implementation and integration of this service with our applications would be categorized as an outcome. We had to share this organizational perspective with the vendor management.

4. FR-Agile required substantial changes to the methods of operation, when compared with Agile. Vendor teams were not provided complete transparency into the Cyber Security space and critical application data, owing to compliance requirements. It was very essential to establish mutual trust with the vendor team and explain the reasons of this exception.
5. Differences of opinion between the client and vendor teams almost often arise in larger initiatives and must be resolved swiftly and amicably. The central idea that we all win when we succeed must be at the heart of every contributor.
6. When using a SOW (Statement of Work), try to be creative by incorporating all the information on the engagement, team structures, exceptions on the methodology used in the company and the prerequisites that must be met before the vendor can deliver work.

Five weeks after the vendor staff was fully onboard with the client team and the expected deliverables, we started making solid progress.

Conclusion

The Public Cloud Platform MVP was successful in meeting its objective through the effective use of FR-Agile. The MVP scope was delivered on time. To summarize – *when strategic innovative opportunities requiring fast execution face risks due to the bureaucratic processes, organizations can look at empowering smaller teams to enable change. These special-purpose, interim teams can leverage the nimbleness, empiricism and innovation that a truly Agile enterprise would otherwise provide, within a traditional organization that is not fully transformed as Agile.* The forming of a team charter, identifying risks, adopting a set of collaborative measures to mitigate the known risks, and working iteratively with more independence will enable the staff to achieve their important objectives. The special-purpose teams can be dismantled once the tactical objectives are met.

For any FR-Agile effort to be successful, all the stakeholders must be fully committed to the decisions. Capturing team decisions on methods of collaboration, tracking requirements and reviewing all the information periodically within the team will boost the performance of such teams. Providing guidelines on the specific atypical procedures that a team follows will also furnish any new members and vendor staff with useful guidelines without any significant loss of productivity.

With the advancement of Artificial Intelligence, Cloud Computing and Big Data, more product development will revolve around the creation of prototypes in the decade. The functional prototypes will decide on whether a corporation will move forward with the investment of a full-scale development. *By using Fitted Requisite-Agile even the most mature corporations, employing a traditional outlook, can eliminate the delays and risks associated with a process-centric approach and achieve their targets of innovation.*

About the Author



Chaitali Chheda

North Texas, USA



Chaitali Dinesh Chheda is a technology leader and author with experience in technology, management, financial services and coaching. In her 15-year career, she has led, trained and managed individuals and programs in her corporate role as well as independently. She's presenting at the UTD Symposium for the third time. Her paper from 2019 was published in the PM World Journal in September of the same year. Apart from technology consulting, she enjoys cooking, traveling, reading, tennis, coaching the young and exploring new opportunities. She is based in the Dallas / Fort-worth metroplex in the United States.

She can be contacted at chaitalichheda@gmail.com