The Evolution of Agile in Modern Software Engineering

01331388

***Abstract*—Many consider Agile to be the definitive, cut-and-paste solution to the issues plaguing their organisation, yet organisations can do more harm than good when attempting to force Agile where it is not required.**

I. Introduction

This article explores the widespread adoption of Agile and it's influence on team structure, size, scalability, resilience and it's impact on the overall tech industry. The aim is to investigate several companies that are exploiting and manipulating Agile to their advantage particularly those operating outside of the standard Agile framework. This article explores the evolution of the software engineer and introduces several case studies where organisations are challenging orthodox executions of the role.

II. The Demand For Agile

The Manifesto for Agile Software Development was published in 2001 by a series of well-known developers, founders and technology experts [1]. These engineers highlighted several issues they encountered while working with their current development systems, the main system being the Waterfall Model. The Waterfall Model follows a chronological plan with distinct sequential phases; teams operate independently and in isolation, with a primary focus on requirements and outcomes due in fixed dates, yet clients are not queried or considered in the design and implementation stages resulting in an inferior final deliverable [2].

Waterfall teams typically include more than 15 people due to their hierarchical structure [2]. Teams have 4 roles: a developer, a tester, a business analyst and the project manager. These roles are stressful due to the fragility of the waterfall system; developers work hard to ensure their final product is market-ready to avoid compromising the project timeline, testers are tasked with finding all the issues with the products and reporting these back to the developers, business analysts must market the products and provide business insights to other members. The project manager exists at the top of the hierarchy to which every other team member must report to, they are the only member of the team responsible for assigning tasks and high-level decision making.

Agile offers several solutions to this methodology by introducing an iterative approach to development and reducing team size to a maximum of ten [1]. This iterative approach, compared to the waterfall method's chronological one, significantly reduces time to market as well as increasing flexibility. This has been reflected in every Agile survey since it's inception, with the most recent survey expressing that 64% of respondents experienced faster time to market and 70% highlighting flexibility to changing priorities as a positive change [3].

Since its emergence in 2001, Agile has evolved significantly. At one point extreme programming (XP) was a popular Agile methodology, yet today it is used by less than 1% of organisations [3]. Scrum, on the other hand, has cemented itself as the most popular Agile practice being used by 66% of respondents [3].

Agile is still rapidly being adopted within software development teams with adoption increasing to 86% in 2021 and 97% of companies adopting Agile practices in at least one team [3].

III. Issues with Modern-Day Agile

Agile, in it's infancy, was primarily adopted by enthusiasts and the processes propagated at the grassroots level, inconspicuously or reluctantly approved of by leadership [4].

As Agile grew in popularity, many companies began adopting Agile from the top down. Forcing teams to pick up a new way of working, regardless of their individual circumstances, can lead to internal disputes and a decrease in productivity and team morale hence creating resentment at lower levels. Upper management are often stringent on enforcing Agile, viewing it as the conclusive solution to problems encountered in their organisations.

Likewise, many of the barriers to suitable Agile adoption expressed by respondents (inconsistencies in processes and practices (46%), cultural clashes (43%), general organizational resistance to change (46%)) are a result of upper level mismanagement or skepticism [3]. The first barrier clearly stems from the upper management as managing several teams practicing their own version of Agile is a difficult task especially in a hierarchical organisation.

It is apparent that through focusing on delivering value early and often, as opposed to adhering to an intransigent plan, organisations are able to react quicker and more readily to unforeseen circumstances. Ironically this very principle has been sullied, with many organisations now focusing on implementing Agile strictly as opposed to allowing freedom of decision-making and fluidity within teams.

This was the case was during my placement at Shell, we were required to conduct meticulous code-reviews in compliance with the appointed Scrum framework. Had we been able to deviate from this requirement we would have been able to avoid insurmountable PR backlogs and abandonment resulting in a more refined, productive development focus.

As the size of a team increases, the productivity of individual members begins to drop, a phenomenon known as the Ringelmann effect [5]. Larger teams tend to discourage ownership and participation and encourage loafing. Developers will often feel overwhelmed and unaccounted for, resulting in their decreased participating and productivity. Shrinking of responsibility at an individual level leads to a responsibility imbalance at the group level, the entire business suffers as does every person involved.

In accordance with the team's requests and desires, bloated teams should be split appropriately to ensure that the Ringelmann effect is minimised and responsibilities are correctly balanced [6].

IV. Palantir's Take on Agile

Palantir [7] is a software organisation that specialises in creating software which empowers organisations to effectively integrate their data, decisions and operations.

Palantir's product development organisation is organized in three main groups: Gotham [8], Foundry [9] and the final group for deployment infrastructure; each comprised of a large number of micro-service-shaped components developed by distinct teams [10].

Teams are typically small with four or five members occupying a range of roles: designer, developer, product manager etc. Teams are allowed to self-organize around customer requirements instead of adhering to top-down mandates or implementation plans; while being flat and decentralised to allow each member to make project-critical decisions encouraging autonomy and ownership [4]. Many developers at Palantir enjoy the freedom and autonomy expressing a greater sense of fulfillment compared to more hierarchical and centralised organisations they have belonged to in the past [11].

Two-week sprints, daily stand-ups and retrospectives are commonplace despite the level of independence Palantir offer their employees. These practices are freely modified to suit independent teams. Robert, an ex-team lead at Palantir, distinguishes his team's retrospectives from the traditional definition, focusing on team chemistry and productivity, motivations and threats as opposed to discussing concrete productivity milestones such as features implemented and bugs fixed [10].

Psychological safety, a term coined by professor Amy Edmonson [12], is used to quantify the belief that employees will not be punished or humiliated for making mistakes or proposing ideas, is paramount at Palantir. Several incentives are in place to ensure that employee mental well-being is high and psychological safety is not compromised including the self-explanatory "Take-What-You-Need Time Off" policy [13]. This strategy obviously benefits Palantir as employees must be well-rested and focused, their decentralised yet cohesive system allows for employees to take time off without bottlenecking the project workflow.

Palantir emphasises the communication and articulation between the software engineers in their business and product development organisations. The Agile manifesto emphasises that "Business people and developers must work together daily throughout the project" [14]; Palantir perfectly encapsulates this by enforcing a flat team structure. Removing the hierarchy that would slow down and complicate communication between teams ensures that engineers from the respective teams work extremely closely creating a short feedback loop.

This works seamlessly in conjunction with the first Agile principle: "to satisfy the customer through early and continuous delivery of valuable software." [14]. Palantir achieves this by encouraging forward deployed teams to test their progress with users every day. To inspire efficiency in all aspects, developers are encouraged to evaluate the necessity of meetings and even decline them to minimise wasted time.

The Agile manifesto encourages organisations to "Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done." [14]. Palantir is able to fulfill this commandment early by encouraging new employees to solve complex problems independently upon hire [11]. Although this may be daunting to some and viewed as a sink or swim situation, Palantir believes it is the best way to instill autonomy and ownership. Several team leads have highlighted that these new "Palantirians" with minimal professional experience were pivotal in making customer relationships successful and inducing non-linear change [10]. To contrast, a graduate engineer at Bloomberg stated that he had to endure several months of monotonic training and culture onboarding before progressing to an actual development team, although onboarding is a great process for integrating new hires it can often lead to lower levels of motivation and enthusiasm if drawn-out.

Roles are fluid within teams at Palantir, whereas roles are commonly static in Agile. Carlene, a business SRO at Palantir, describes how a Deployment Strategists was both developing the software and acting as a joint product owner with their business partner in the customer team and how this initiative was critical in the final product deployment [15]. She emphasises that formulaic interpretations of Scrum or other versions of Agile are resource-expensive due to their specificity of roles and implementations. She states "In many ways these can be contradictory to the core principles of Agile that champion adaptability in response to evolving constraints" [15]. It is important to not fall into the trap of quantifying progress and team satisfaction through their ability to be Agile but rather through the quality of their deliverables and client satisfaction. Often this is only possible by operating outside of the traditional Agile constraints.

V. The Forward Deployed Software Engineer

Modern-day software engineers are responsible for the development and maintenance of software products and rarely have an input in the prior product planning and discussion stages i.e. determining the features a particular product needs. Organisations intentionally divide the production pipeline, separating the teams and responsibilities to allow for easier management, abstraction and compartmentalisation.

Unfortunately this compartmentalisation is anti-Agile; while many think that Agile primarily focuses on cohesiveness

within the team, synergy and communication between teams is extremely important. Limiting a software engineer's scope of influence can lead to a lack of fulfillment as they are unable to contribute meaningfully to the design phase nor are they able to directly interact with the consumer or client post-deployment. Instead they rely on the customer facing team to relay information to the designers and product managers who, in-turn, relay the information back to them. It can take at least 2 stages of communication before the information reaches the software engineer, this propagation is inefficient and can lead to issues with miscommunication.

Palantir attempts to remedy this disconnect by introducing an extremely important and unorthodox role, the Forward Deployed Software Engineer (FDSE) [16]. FDSEs are an integral part of Palantir's business department. The FDSE operates autonomously across the entire product pipeline to deliver customised solutions, iteratively, one client at a time. The process often begins with an open ended question such as "How can we better identify instances of money laundering?", the FDSE must either determine how to leverage existing Palantir solutions to address their issue or work with the Product Development teams to produce a new solution [11].

FDSEs engage in the typical software engineering practices, customised to their discretion: engineering reviews, code reviews, deployability optimization, maintenance and monitoring of production systems. They are also tasked with relaying aspects of their development to the internal development teams, both business and product. Although the features FDSEs produce are intended for a single client, working with the business and product development teams helps to identify where the product can benefit other clients and support in the generalisation of the product.

Palantir explicitly advertises that FDSEs work in small teams with minimal supervision and highlights their unique involvement in the end-to-end execution of important projects. [17]. This is to attract technical candidates eager to take responsibility and autonomously work in a rapidly changing iterative environment that would otherwise only be possible at a startup in a position of seniority, to the extent that the FDSE job advert compares the responsibilities of the role to those of a startup CTO [17].

FDSEs employed at Palantir stress that it is vital to abandon any preconceptions on the role to adopt an entirely new perspective of engineering [11]. One in particular had previous experience as a startup CTO, reiterating the similarities between the positions as highlighted in the FDSE job ad. Another FDSE enjoys the rapid cycle between creating solutions and seeing them in action, delivering impactful solutions to client in short space of time and the unpredictability, that encourages constant learning and versatility [16]. He highlights a specific question that perfectly encapsulates the technical, social aspects of the role: "How do I design, build, test, deploy and maintain a unique workflow to allow a particular non-technical customer to visualize and interact with high-noise data? How can I generalize this feature to fold it into the base platform, so that other FDSEs and clients can benefit from my work?" [16].

It is clear that the platforms of existing solutions, small team sizes and allocation of one team per client are significant contributors to why the FDSE role works so well in Palantir. However, adoption of the Forward-Deployed Engineer role could prove useful in other organisations with customer-facing engineering departments. Forward-deployed engineer positions at corporations other than Palantir are beginning to surface as organisations recognise the value in the position [18], [19]. The FDSE has the potential to dramatically alter the entire software engineering landscape, acting as a bridge to software engineering for more business-oriented talent, especially from fields such as consulting. The popularisation of this role will be extremely beneficial to Palantir as they are the pioneers and can influence and determine the direction and definition, and have a first-mover advantage when it comes to hiring and talent acquisition. The role can help combat the aforementioned compartmentalisation and rigidness creeping into present-day software engineering organisations.

VI. Incentivising Innovation

An issue with Agile is that it seems to quell seemingly inefficient entrepreneurship within organisations due to the insistence that teams focus on creating products that work. At Palantir, designers are encouraged to work on their own projects with one week per year dedicated to an individual passion project, Hack Week [20]. Blueprint began as a single designer's solution to the disconnected nature of their current ecosystem of applications. Palantir identified the value in the side project and supported the designer to continue working on the system, staffing the project with a team of engineers. The ability to identify and prioritise worthwhile projects as well as allocate teams to said projects is made possible by Agile and the iterative development cycle.

Palantir was adamant that specific principles were at the forefront of development: composition, reusability, accessibility, developer experience, while allowing the team design freedom. Blueprint has since emerged as a popular open-source React-based UI web toolkit used by 10,635 developers and advanced by 294 contributors [21]. Many developers praise the platform, specifically its modularity, customisability and scalability [22].

Palantir incentivise their developers to experiment with newer risky technologies, taking an elevated level of risk for a potentially more rewarding payoff related to actually shaping and influencing the new technology, advancing their tech stack and attracting higher quality candidates. In particular they allowed Blueprint to be built using TypeScript, a relatively obscure, underrated JavaScript tool at the time, which has grown to be massive and the popular option among JavaScript developers. TypeScript was instrumental for providing a superior developer experience amassing adoption within the organisation.

VII. Scaling and Hiring

Palantir needs to scale, desperately. Several correspondents have relayed that their teams are currently overworked due

to an influx of new clients. One FDSE in particular highlights that his team of six is responsible for four projects from one of Palantir's biggest clients. He mentions that his team lead is currently overloaded managing the projects, insuring critical objectives are met.

Senior management recently addressed this issue in an internal panel, expressing that their top priority is to hire and to scale, explicitly stating they wish double their workforce. Such a dramatic change must be intensely managed and a exhaustive and resilient framework must be in place to hire and accommodate new employees. The current proposal is to scale teams both horizontally and vertically, increasing team size to an average of six and increasing the number of teams to balance the project load, enabling them to deliver a more personalised and complete solution to their clients.

To reiterate, the problem is not with the platform products, which are easily extensible, but with the increased demand in these products. The perfect solution to this is to blitzscale [23], especially considering Palantir does not need to compromise on their products, nor be cautious about potential resilience and reliability issues.

To achieve their scaling goals Palantir has advertised a five-figure referral bonus to incentivise existing employees to assist in the hiring process and attempt to mitigate pressure from their recruiters. This is similar to a hiring initiative at Uber during their blitzscaling process where managers would ask newly hired engineers about the best three engineers at their previous job and sending said engineers direct offer letters, bypassing the costly interview and reference checking processes [23].

Palantir acknowledges that people are their greatest resource, that every role in their ecosystem is crucial to delivering comprehensive solutions to their clients. However talented people are difficult to hire and Palantir's reputation may be a negative factor in hiring. Palantir has a rigorous hiring process, with six separate technical interview stages for their graduate positions spanning 4-6 weeks [24]. These hiring processes are expensive but Palantir deems them necessary to employ the right people at their corporation, both technically and culturally. However there comes a time when every corporation must compromise on certain aspects of their culture in order to scale to meet demand and progress at the forefront of the industry, or risk losing market share to competitors.

A solution to the hiring problem may be to lower requirements and allow for an increase in junior developers. An engineer at Palantir agrees with this sentiment, arguing that Palantir should focus on improving existing documentation to allow for new hires to easily pick up and understand products offered across their platforms instead of training and hiring senior developers that are able to pick it up faster. This decision would result in a drastic reduction in hiring expenses as senior engineers are paid almost double that of a junior engineer. This strategy was employed to scale the data science department at Shell, we spent several weeks documenting frameworks that we were comfortable using consequently allowing new hires to comfortably traverse said frameworks without disrupting other engineers. This proved to be successful as we fluidly onboarded several data scientists and engineers using the documentation and witnessed a drastic reduction in the time taken for the new hires to begin contributing, from two weeks to three days.

Another unexpected issue threatening their plan to scale relates to the salaries demanded by their engineers. Employees are currently vocal on the social media network, Blind, about their dismay at the proposed compensation changes. Having a significant number of employees threatening to leave at a time during which you are hoping to scale is, to say the least, terrible. Palantir engineers expect an industry leading salary and compensation benefits due to the demanding and highly technical nature of their work. This demand must be met in order to retain experienced valuable engineers who will prove crucial in a time of intense scaling. The cost in retaining the engineers must be weighed against the cost of hiring top level engineers and this additional burden on the hiring department must be accounted for.

VIII. Agile and Resilience

Agility and resilience are coincident, one can not be achieved without the other, as demonstrated brutally by the Covid-19 pandemic. The technology sector thrived while others faltered, an outcome clearly reflected in the stock market with the NASDAQ-100 outperforming every other sector.

At Palantir, Agile is an important inspiration for resilience. Employees within a team can span multiple time zones and with a recent rally for remote working, operational resilience, stability and team cohesiveness are as important as ever [25].

Threats to operational resilience do not necessarily stem from poor leadership or neglectful crisis response planning. Often a disconnected system architecture and a lack of thorough system maintenance and testing, coupled with assumptions that reflect a simpler world, leads to organisational fragility. Often a crisis is the trigger, that encourages teams to acknowledge their rigidness and fragility, forcing them to make the necessary changes and compromises to adopt a more agile approach to working or risk failing altogether [26].

An insightful way to cultivate resilience is by encouraging every team member to actively participate in decision making, as Palantir does. This ensures that team members are dynamic and enterprising, and motivated to resolve issues.

IX. Conclusion

To conclude, we have explored several components of the Agile ethos and discussed the benefits and negatives of both conforming to Agile and operating outside of it. The overarching message is that Agile should be adapted to suit your organisation; it is not a concrete doctrine. Especially considering industry-defining change comes from those innovating unconventionally.

Acknowledgments

A special thank you to the Palantir engineers, that wished to remain anonymous, for assisting me in my research.

References

[1] K. B. et al., "Manifesto for agile software development," Available at https://agilemanifesto.org/.

[2] Workfront, "Waterfall methodology," Available at https://www.workfront.com/en-gb/project-management/methodologies/waterfall.

[3] digital.ai, "15th annual state of agile report," Available at https://digital.ai/resource-center/analyst-reports/state-of-agile-report.

[4] V. S, "Has agile gone rigid? (how we do agile, 2)," Available at https://blog.palantir.com/how-we-do-agile-2-70032b3bffd.

[5] T. van Schneider, "The psychological theory that explains why you're better off working solo," Available at https://qz.com/848267/the-ringelmann-effect-productivity-increases-when-youre-working-solo-rather-than-on-a-team/.

[6] D. Van Heerden, "Splitting software delivery teams," Available at https://www.linkedin.com/pulse/splitting-software-delivery-teams-daniel-van-heerden/.

[7] Palantir, "Palantir homepage," Available at https://www.palantir.com/uk/.

[8] ——, "Gotham: The operating system for global decision making," Available at https://www.palantir.com/palantir-gotham/.

[9] ——, "Foundry: The operating system for the modern enterprise," Available at https://www.palantir.com/platforms/foundry/.

[10] R. Fink, "How we do agile," Available at https://blog.palantir.com/how-we-do-agile-1-35fa76e48cad.

[11] C. E, "Who wants to be a delta?" Available at https://blog.palantir.com/who-wants-to-be-a-delta-8d2ea948035.

[12] A. Edmonson, "Building a psychologically safe workplace | tedx talks," Available at https://www.youtube.com/watch?v=LhoLuui9gX8.

[13] Palantir, "Life at palantir," Available at https://www.palantir.com/careers/life-at-palantir/.

[14] K. B. et al., "Principles behind the agile manifesto," Available at https://agilemanifesto.org/principles.html.

[15] C. K, "Agile1 + agile2 = ? (how we do agile, 3)," Available at https://blog.palantir.com/how-we-do-agile-3-dd8a2414cf4e.

[16] Palantir, "A day in the life of a palantir forward deployed software engineer," Available at https://blog.palantir.com/a-day-in-the-life-of-a-palantir-forward-deployed-software-engineer-45ef2de257b1.

[17] ——, "Forward deployed software engineer job advert," Available at https://jobs.lever.co/palantir/5168e8fd-fec1-4fea-b7a1-81bdaea65850.

[18] R. Health, "The role of forward deployed engineers at ribbon and beyond," Available at https://www.ribbonhealth.com/blog/forward-deployed-engineers-at-ribbon/.

[19] R. Walters, "Forward deployed engineer: Job details," Available at https://www.robertwalters.co.uk/informationtechnology/jobs/softwareengineering/1516074-forward-deployed-engineer.html.

[20] Palantir, "Break the machine: Hack week 2021," Available at https://medium.com/palantir/break-the-machine-hack-week-2021-990164631be2.

[21] ——, "Palantir blueprint github repository," Available at https://github.com/palantir/blueprint.

[22] iwangotamarjo, "Blueprint.js, a gift from palantir to the open-source community," Available at https://www.reddit.com/r/PLTR/comments/o6pd2g/blueprintjs_a_gift_from_palantir_to_the/.

[23] T. Sullivan, "Blitzscaling," Available at https://hbr.org/2016/04/blitzscaling.

[24] Palantir, "Getting hired," Available at https://www.palantir.com/careers/getting-hired/.

[25] E. Nguyen, "From fragile to resilient: Building a connected company," Available at https://blog.palantir.com/from-fragile-to-resilient-building-a-connected-company-3a4f98a6518a.

[26] E. Chong, "Agile resilience in the uk: Lessons from covid-19 for the 'next normal'," Available at https://www.mckinsey.com/business-functions/people-and-organizational-performance/our-insights/agile-resilience-in-the-uk-lessons-from-covid-19-for-the-next-normal.