

Rocket Mortgage Delivers Twice the Value at Half the Cost at Scale

David Juan David Negron Michael Simmons Jeff Sutherland
Rocket Mortgage Rocket Mortgage Scrum Inc. Scrum Inc.
davidjuan@rocketmortgage.com davidnegron@rocketmortgage.com michael.simmons@scruminc.com jeff@scruminc.com

Abstract

The conventional wisdom of many Agilists is that Scaled Agile and Scaled Scrum are incompatible. This was tested in 2018 when Rocket Mortgage used a Scaled Agile model to organize 2,000 of their 26,000 team members into teams, and the teams into collections of "release trains" centered around business value streams. The Client Marketing Release Train then took the Scaled Agile model that all release trains were implementing and layered Scaled Scrum with DevOps practices on top of it. By doing so, they reduced the average feature cycle time from 83.7 days to 11.6 days and increased feature delivery by 721% with higher quality. Here we describe the tools and techniques they used to deliver more than twice the value at half the cost.

1. Introduction

Rocket Mortgage wanted to scale Agile while still providing autonomy for engineering teams. Agile environments are critical to adapting to changing conditions, and during COVID-19, Agile business units outperformed their non-Agile counterparts by up to 94% [1]. Rocket Mortgage began formally adopting a consistent Agile methodology in Q3 of 2018 before COVID-19 by implementing a Scaled Agile framework. By Q4, the implementation resulted in the formation of 26 release trains and improved feature throughput by 101% by Q3 2019, as measured by the number of features completed in a program increment. Deployment continues to expand, and in 2021 there are 41 release trains of roughly 30-90 people each.

But MIT Sloan Business Review [2] reports only 17% of leading companies today will remain leaders in five years. Knowing this, the Rocket Mortgage Director of Engineering for the front end of the mortgage loan systems began evaluating the Scaled Scrum framework as a tool for improving their Scaled Agile implementation. This is significant because it is commonly assumed that Scaled Agile and Scaled Scrum are incompatible and are either/or options. Here we address the results when Scaled Scrum is added

correctly to a highly successful Scaled Agile implementation.

2. Comparable Industry Case Studies

With the multitude of benefits that a properly implemented Scaled Agile process delivers, the reasons for undergoing a Scaled Agile transformation can vary significantly from one implementation to the next. However, research shows the primary driver for moving to Scaled Agile, when surveyed across implementations at the organization, division, and project levels, was to reduce time-to-market [3].

Scaled Agile implementations can often cut cycle times in half. Cerno, a leading software development company in China, experienced a 58% reduction in cycle time from their Scaled Agile implementation [4]. Another company, EdgeVerve, whose industry-leading financial software is used by banks in 94 countries, implemented Scaled Agile that resulted in a 50% cycle time reduction [5]. Royal Philips, the medical technology company, reduced cycle time by more than 58% through Scaled Agile [6].

However, as more people and organizations learn of the benefits of Scrum, its implementation is growing from just teams up to divisions and organizations. A systematic literature review found that what was experienced as a lack of communication between teams was caused by Scrum not being implemented at a system level. As one Scrum Master (SM) put it, "We had Scrum within our small groups, that's about it [7]". This realization of the benefits of Scrum at an organizational level over that of a team or division level is a significant force driving Scaled Scrum implementations.

3. Research Methodology

Takeuchi and Nonaka reviewed the best lean hardware teams worldwide and published the first paper introducing Scrum Project Management in 1986 [8]. Sutherland created Scrum for software development in 1993, has worked with Schwaber since 1995 to create the Scrum Guide [9], and created the Scrum@Scale

Guide in 2016 [10]. Scrum is rooted in lean, and the authors have worked with Takeuchi and Nonaka directly since 2011 and published a second Harvard Business Review paper on Scrum [11]. The authors are coaches and trainers for Toyota in Japan, the United States, and Europe and worked closely with the Lean Enterprise Institute (LEI) to incorporate lean tooling into Scrum. John Shook, a former CEO of LEI, created the first Toyota plant in the United States [12] and helped proliferate the A3 Process [13, 14].

The A3 Process, used as the research methodology for this paper, is fundamental to process improvement at Toyota. Its evolution began at the end of World War II when General MacArthur brought Fundamentals of Industrial Management to Japan [15], followed by W. Edwards Deming and others in the 1950s [16]. Taiichi Ohno developed this into the Toyota Production System [17]. Sutherland worked with many A3 experts at Toyota, including Mike Tromas at the Toyota Kentucky plant, who used A3 to introduce Scrum into assembly-line production support and more than doubled productivity at scale [18].

The research methodology of this paper uses the six-step A3 process to describe the background, current condition, target for improvement, root causes analysis of major problems, recommend interventions, and follow-up in a way that other organizations can implement the recommended changes [19].

4. Rocket Mortgage Background

Before a Scaled Agile organizational change, the technology teams at Rocket Mortgage were organized into roughly eight technology-centered "platforms," with each group responsible for large pieces of the underlying technology. Each platform was staffed by a group of teams with almost total autonomy in how they operated. As a result, there was a range of development methodologies from traditional project management to Kanban.

There was a need to realign the ownership of projects and initiatives across the organization. In July 2018, the overall portfolio consisted of more than 275 "high-priority" work items ranging in size from small projects to large-scale initiatives. These were prioritized quarterly by a large group of business and platform leaders and roughly coordinated by project managers and "Epic Owners." Additionally, platforms and their component teams managed individual work backlogs fed by requests from multiple business areas and partners across the organization. There was little structured coordination from top to bottom. As work was divided up and funneled down to teams, the priorities, processes, and working structures became

progressively independent and disparate from one another.

A Scaled Agile transition organized teams into collections of release trains centered around business value streams with a common Scaled Agile methodology. These groups of release trains, called "streams," were strategically aligned around similar business and technology capabilities, explicitly covering the mission-critical areas of Product Engineering, Data Intelligence, Infrastructure and Operations, Security, and Enterprise Services. This restructuring happened all at once: the organization shifted from technology capability-centered platforms in the third quarter of 2018 to 23 value-focused release trains in the fourth quarter of 2018 (42 as of April 2021). The Scaled Agile restructure allowed teams to plan, commit, and execute together, communicating in a unified Agile language and adhering to standardized processes. Technology teams at Rocket Mortgage were becoming more focused on delivering value for common business and technology missions.

The Scaled Agile transition also included rolling out several tools and measurement guidelines and provided training and coaching resources. All leaders from the technology and product organization, along with many team members and business leaders, were trained in Scaled Agile and development lifecycle practices over thirty days during the third quarter of 2018. Each release train created a DevOps roadmap, tracking their progress on the DevOps wheel [20]. Several large-scale initiatives introduced standardized methods and pipelines into the software development lifecycle. Internal tools were also created leveraging integrations with the company's primary work-tracking tools and rolled out to provide visibility into progress and establish consistent productivity measurements.

After restructuring into release trains, while the epic portfolio remained, it was trimmed from 275 epics in July 2018 to just 37 by October 2019, reflecting only large solution (multi-release train) initiatives. Epics were prioritized through a continuous refinement process by a handful of leaders, with ownership of feature-and team-level work transferred to the release trains, with the ultimate goal of empowering self-sufficient release trains and autonomous value delivery.

The Scaled Agile rollout drastically improved feature delivery time and predictability across the organization by a factor of 2x. Feature Cycle Time was cut by more than half, from 71 days down to 33 days. The company also improved cycle time standard deviation by 49%, from 82 days down to 42 days. As a result, when normalized for a 10-week release timebox, feature throughput improved by 101% from 414 features delivered per release to 833 per release. In addition to producing more incremental value across the

organization, the Scaled-Agile implementation was the catalyst for increased commitment completion percentages. The planned work completed during a program increment increased by 37%, from 60% in 2017 to 88% in 2019.

With Rocket Mortgage's own Scaled Agile implementation resulting in a 46.5% reduction in cycle time, the Client Marketing Release Train was already looking for something better.

5. Client Marketing Improvement Opportunities

Client Marketing provides the client-facing frontend of the mortgage system with high demand for new and improved functionality, delivering a better user experience. As the leadership looked at the next steps in improving their Agile implementation, it became clear that four areas needed upgrading:

1. Client Marketing Scrum teams were using inconsistent tools and techniques. All teams needed better ways to integrate work at higher velocity.
2. There were many different specialized roles on the Scrum teams. Less specialization with all roles focused on delivery could improve results.
3. Communication across all teams needed to be clear, consistent, and more productive.
4. Cycle times needed to be reduced further.

6. Root Cause Analysis

Scaled Scrum enables a performance analysis of components of scaling frameworks using a heat map [21]. Each column represents an organization, and each row is a Scaled Scrum component. Green is great, and red is blocked. Figure 1 is a photo of the actual heatmap used by Client Marketing in their analysis of the items identified in this paper.



Figure 1. Scaled Scrum Performance Heat Map

Some scaling components relate to the teams (team process, team coordination, continuous improvement, delivery), others relate to the Product Owner (PO) (vision, portfolio prioritization, backlog refinement, release planning). Key components relate to the entire organization (Executive Action Team, MetaScrum, product release, metrics) [10]. By evaluating the effectiveness of each component, a prioritized list of improvement initiatives can be generated. This list of improvements is then driven by an Executive Action Team that runs like a Scrum team. Client Marketing quickly identified several targeted areas for improvement using this process. Improvements are prioritized by maximum impact for minimum effort and reprioritized after every individual improvement implementation. The system is constantly changing, so the A3 Process is a highly effective way to target the prioritized dysfunction's root cause.

6.1 Scrum Basics

A performance analysis revealed that inconsistent Scrum implementation was a significant impediment to efficiency. In a review of all Client Marketing Scrum teams, it was found that roughly 50% had significantly altered four (Planning, Review, Daily Scrum, Retrospective) of the five Scrum events. Additionally, there was no uniform application of the three Scrum roles (PO, SM, Team) across Client Marketing.

A related issue discovered was the existence of too many roles. Based on the work of the Pasteur Project at Bell Labs with over 200 published case studies, there was ample evidence that too many roles caused poor communication saturation increasing the need for meetings and extensive rework. This poor communication environment was the primary driver of reduced velocity [22]. Leadership hypothesized that an abundance of different roles within Client Marketing was causing process constraints, hindering communication, and slowing down work.

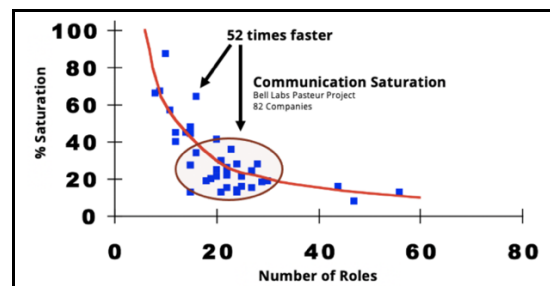


Figure 2. Roles decrease communication saturation

As seen in Figure 2 from the Bell Labs Pasteur Project [23] involving the first 82 companies audited,

communication saturation drops as roles are added. The circled area represents most companies with 20+ roles at ~25% communication saturation. Thus, fewer roles leads to improved communication among team and organization members. For Client Marketing, this directly impacted the target condition of all functions being focused on delivery.

6.2 Deployments Too Far Apart

Inconsistent structures, tools, and techniques on Scrum teams were delaying deployments. A recent 7-year study with 13 preparatory case studies developed hypotheses to be tested using an academic framework for understanding what makes Scrum teams effective [24]. Findings indicate that two primary variables determine Scrum team results – frequent releases and a clear understanding of stakeholders' needs. At Client Marketing, deployments occurred only bi-weekly (after each Sprint), took on average two hours, and had to be done after 10 PM local time. These restrictions increased batch sizes and caused unacceptable delays in time-to-market for product enhancements, bug fixes, and new product deployments.

6.3 Testing

Analysis revealed that testing was a bottleneck, a problem often encountered in software development environments. Time to test and fix in a later Sprint compared to the current Sprint can be 24 times longer for complex hardware/software projects. This has been observed in Europe and Silicon Valley. A recent example was an investigation at Simplivity, a cloud infrastructure company that is now part of Hewlett Packard [25]. For a technology company like Rocket Mortgage, automation of testing and deployment is a good solution for this problem [26]. A small Scaled Agile implementation cut delivery time by 75% with an exemplary DevOps implementation [27].

6.4 Meeting Structures Did Not Drive Delivery

Scrum is designed to minimize meetings and reports as these are the largest source of waste in most organizations. They significantly reduce process efficiency [28], which radically reduces throughput. Systematic, a large consultancy in Europe operating at CMMI Level 5, implemented Scrum and cut project costs in half [29]. Client Marketing leadership did an ROI analysis of all meetings, including the time the meeting took, topics discussed, attendees, and value of outputs and reports, and decided to eliminate anything non-essential to the Scrum framework [11].

It was found that priorities and outcomes were not consistently communicated across teams. They were often discussed in separate, loosely attended (often-canceled) meetings with marketing leadership primarily focused on lists of deliverables and delivery dates. Meanwhile, leadership leaned on individual meetings with a single Product Manager and monthly sync-ups with each engineering team to prioritize multiple value streams. These delivery teams planned work quarterly with the organization and experimented inconsistently with shorter cycles to address uncertainty and wasted effort.

A daily "Leadership Standup," which followed the typical pattern of a status meeting, was the only organized daily communication. Each SM would talk about what they would be focused on for the day, with updates typically consisting of a list of meetings and action items receiving their attention. This meeting took place first thing in the morning, before the SMs had their Daily Scrum with their teams, produced little to no value, and was disparate from what each team was doing in delivering value or being hindered by impediments.

7. Countermeasures Taken

Because Scaled Scrum and Scaled Agile are both major Agile frameworks, it is generally accepted that one or the other should be selected [30]. But while the rest of Product Engineering stopped at Scaled Agile, Client Marketing took the extra steps of putting Scaled Scrum on top of Scaled Agile. This allowed them to address all the issues found in their root cause analysis and reach their target conditions, essentially applying the A3 Process to Scaled Agile. As part of the process of adding Scaled Scrum, the countermeasures below were put into place.

7.1 Scrum Reset

In addition to the organization-wide Scaled-Agile transformation, Client Marketing employed four primary tactics to improve and scale its Scrum practice. First, Client Marketing senior leadership completed training with Jeff Sutherland on scaling Scrum. Second, Client Marketing was retrained in the three Scrum roles, the five events, and the three artifacts. Third, all of Client Marketing did a "hard reset," establishing a strict implementation of Scrum at the team level. While each team within the release train was already doing some variation of Scrum, leadership set the expectation that each team would strip away any complexity that had been layered on over the years and revert to following the Scrum Guide [9].

Before the reset, only one-third of Scrum Teams held Daily Scrums, Sprint Planning, Sprint

Retrospectives, or had a Definition of Ready. None of the Scrum Teams had a Definition of Done. After the reset, all teams followed each ceremony as defined in the Scrum Guide [20] and had Definitions of Ready and Done. Finally, the Client Marketing leadership began to operate as a Scrum team as well.

7.2 Scrum Roles

Consistent with the Scrum Guide [9], it was agreed that POs were primarily responsible for increasing the product's value, while SMs were accountable for accelerating delivery. All permutations of these roles were eliminated.

7.3 MetaScrum

The MetaScrum is a regular meeting (at least once a Sprint) with Senior Leadership, the Chief PO, and senior members of the PO and engineering teams [10]. The purpose is to align the organization with the Chief PO's backlog. This meeting was first implemented at PatientKeeper [31] in 2003 and was later formally defined by the Scrum Patterns Group [32] as essential to high-performing organizations with many Scrum teams.

Client Marketing implemented a weekly MetaScrum to coordinate prioritization and dependencies and achieve ongoing alignment with stakeholders. The process was led by the Release Train Engineer and involved consolidating all prioritization, train-level work intake, feature refinement, and dependency coordination discussions down to a weekly meeting with all stakeholders, product managers, and leaders present. This allowed prioritization to be aligned through POs down to each team.

7.4 Scaled Daily Scrum

As MetaScrums began, Client Marketing leadership started holding a Scaled Daily Scrum (SDS) with all the SMs in the Release Train. Two of the most significant changes of the SDS from the previous Leadership Standup were the time it occurred and its focus. The SDS occurred after each team's Daily Scrum, and the focus was purely on delivery. With a dashboard with Sprint burndown charts for each team prominently displayed on a screen, each SM would report on their team's progress towards the Sprint goal and any impediments.

If a team did not clearly show to be trending towards early Sprint completion, the SM was expected to bring one or more impediments to the SDS that they were focused on removing. If the SM could not eliminate an impediment within hours, it became the

highest priority item the Release Train Leader would focus on. The SDS converted a low-value status report meeting about the schedules of SMs into a high-value event focused on taking immediate action to resolve challenges that same day that teams were facing to hit Sprint goals. This Scaled Scrum ceremony, along with the Daily Scrum, directly attack and resolve decision latency. The importance of this cannot be overstressed, as the Standish Group noted, "The value of the interval is greater than the quality of the decision" and "The root cause of poor performance in a software project is slow decision latency [33]."

7.5 Role Alignment

The leadership team examined all roles and their responsibilities. Any role that was not playing a direct part in delivery, or was causing a bottleneck, was eliminated or repurposed. With SMs focused on removing constraints within the team's delivery process, it became increasingly clear that traditional quality assurance and specialized roles that only concentrate on a small part of the delivery lifecycle were causing bottlenecks. SMs worked with team members in these roles to either broaden their skillset to apply to any user story in the backlog or helped them find positions within the broader organization that still required their high degree of specialization.

For team members in the role of Quality Analyst (QAs), there was a six-month transition period during which they became Software Engineers (SEs). Those who were unable to do so were transferred to a specialty test group within the larger organization. Business Analysts (BAs) became Software Engineers, POs, or were assigned to the specialty test group.

7.6 Testing

With the elimination of QA roles, the testing requirements increased for SEs. Because there was already a bottleneck in testing, the Client Marketing leadership team took corrective action. Automation testing was increased. This reduced the bottleneck and minimized the amount of manual testing. Unit, accessibility, and regression testing were all automated. The only testing still performed manually was visual testing on devices and browsers.

7.7 Quality Gates

Quality gates were added to the pipeline to ensure the high standards of Rocket Mortgage were maintained. The quality gates verified that increased performance and shorter time to market were not

detrimental to quality. Successfully passing the gates meant successfully passing performance, regression, and accessibility testing. The gates added were:

Accessibility Testing

An automated testing tool that produces a score value which is then compared against a predefined minimum score to be considered passing.

Secrets Scanner

The Secret Scanner looks for plain text secrets within source code. Anything that could grant knowledge or access to items that people should otherwise not have access would be considered a secret.

Security Scanner

Security Scanner is an automated scan that checks for any dependency vulnerabilities.

Content Scanner

A content scanner looks for language-specific files and directories that should not exist within source control.

Static Code Analysis

A Static Code Analysis tool measures code quality via unit-test coverage and reports information about code quality to CI tools.

7.8 Deployments

The ability for each team to autonomously deploy components on demand has been a game-changer in many high-performing organizations like Spotify [34]. Deployment rates were increased from the end of every Sprint to an on-demand Continuous Integration/Continuous Deployment pipeline (CI/CD). Dependencies on other release teams were eliminated to allow for on-demand independent releases by each team. Deployments could occur at any time due to downtime being eliminated through the use of Blue-Green Deployments [35].

Previously, deployments happened at a maximum of once per week, with a typical cadence of once every two weeks. This was because code deployments involved an external team that manually deployed code to production environments, and this could only be done during late evening hours because of downtime that would occur. The deployments took multiple hours of planning and coordination. The actual code deployment would last 2-3 hours, frequently resulting in needing to roll back the deployment if changes caused unanticipated errors in production.

7.9 Infrastructure

The current infrastructure dependencies caused bottlenecks and increased cycle times. All Scrum Teams were dependent on an external infrastructure team that would manage on-premises servers. The

servers were a black box to the Software Engineers because only a Systems Engineer on the infrastructure team was allowed to access it directly. The Systems Engineers did not sufficiently understand the production code being deployed to these servers because that was the responsibility of the Software Engineer. This often led to hours spent debugging environment issues and constantly having to work towards keeping a consistent state across the testing, staging, and production environments. Software Engineers would develop locally but could not test their code until they promoted it to the test environment, which would lead to situations where code could not be deployed because it was waiting for a test server to become available. Because the test and staging environments were managed separately, there were often differences between the environments, causing additional issues in testing changes before releasing to production.

To eliminate these dependencies, infrastructure was containerized, and a Docker environment was implemented, including automated migration and setup/termination as needed using terraform scripting. Additionally, a blue/green deployment model was implemented to test before moving traffic. This allowed each Scrum Team to work independently, free from being dependent on the external infrastructure team.

7.10 OKRs

Objectives and Key Results [36] were established for all Release Trains. The specific relevant Key Results were:

New-New Code Coverage

- A target coverage rate for automated testing of all new lines of code for either legacy or new applications was established.

Overall Code Coverage

- A target coverage rate for automated testing overall (new and legacy code) was established.

Mean Time To Recovery (MTTR)

- The failure of any individual gate within the CI pipeline must be recovered within 24 hours.

8. Results

As one of the largest Scaled Agile implementations, Rocket Mortgage was more successful than most large implementations, reducing cycle time by 46.5%. The goal of implementing Scaled Scrum on top of Scaled Agile was further reduction in cycle time, higher quality, and improved customer and team satisfaction.

8.1 Cycle Time Reduction

By leveraging the Scaled Scrum practices described above, in addition to the organizational Scaled Agile rollout, Client Marketing was able further to improve feature delivery cycle times, feature throughput, and planned work (commitment) completion rates. Table 1 shows that Client Marketing reduced the average feature cycle time by 75%, from 86 days to 21 days. The first year saw a 51% reduction, from 86 days to 42 days, with another 50% reduction to 21 days the second year. The result was that the implementation of Scaled Scrum brought down cycle time another 55.12% over Scaled Agile alone. Additionally, feature predictability improved by a factor of 3x with a cycle time standard deviation reduction from 46 days to 14 days. When normalized for 10-week releases, feature throughput also increased by 340% for the train. Commitment completion increased by 91% for Client Marketing, bringing it on par with the rest of the organization.

Table 1. Pre- and Post-Scaled Scrum and Agile Metrics

	Rocket Mortgage Pre-Scaled Agile End of Q3, 2017	Rocket Mortgage Post-Scaled Agile End of Q3, 2019	Client Marketing Post-Scaled Scrum End of Q3, 2019
Feature Cycle Time	71 days/83 days ¹	33 days	21 days/11.6 days ¹
Feature Throughput	414 / 5 ²	833	22
Feature σ	82 days	42 days	14 days
Commitment Completion	60% / 46% ³	88%	

1. Feature Cycle Times for Client Marketing were 83 days and 11.6 days

2. Feature throughput for Client Marketing was 5

3. Commitment completion for Client Marketing was 46%

8.2 Improved Administration Efficiency

Through restructuring roles and modifying its approach to software quality, deployment operations, and product ownership, Client Marketing saw a change in its ratio of non-delivery roles (team leaders, business analysts, quality assurance) to production-focused roles (engineers and developers). The overall change went from ~2:1 in Q3 2017 to ~1:2 by Q3 2019, indicating an increased focus on delivering working software rather

than managerial overhead. Figure 3 shows the correlation between cycle time reduction (top chart blue line) and the decrease of admin roles (bottom chart blue line). Section one reveals a negative trend where cycle times and non-delivery roles were both increasing. Section two represents the period of change from admin roles. Section three indicates the onboarding of additional teams and adding PO roles (considered non-delivery roles for these graphs, thus the rise in the bottom chart).

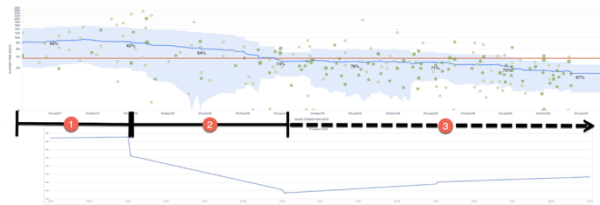


Figure 3. Rocket Mortgage Role Restructuring

8.3 Delivery Focus

The Scrum Reset brought a new level of clarity to the SM and PO. The MetaScrum and SDS provided frequent cycles in which POs and SMs could receive real-time feedback on their performance, take corrective action, and compare results before the next increment. Through this increased visibility and clarity, SMs and POs typically either saw a dramatic improvement in their effectiveness or were able to determine that the role did not fit their strengths objectively. This led to Client Marketing having the right people in key leadership roles, resulting in a high-performance culture focused on delivering value. The result of having effective SMs and POs, along with delivery teams made up of team members with multiple skills, was a team structure and process that drove shorter cycle times for delivering value to clients.

8.4 OKRs

The tracking of OKRs became a bellwether for Rocket Mortgage technical teams to be proactive in their work. Client Marketing's code coverage data is presented in Table 2. Other OKRs either had limited or no pre-Agile data, or contained proprietary information.

Table 2. Client Marketing overall code coverage

2018	2019	2020
43.73%	53.15%	69.34%

8.5 Deployments

Deployment rates were improved through a careful process. After having all prerequisites in place, releases could be done on-demand using a CI/CD gated pipeline as previously mentioned instead of the previous once per Sprint limitation.

8.6 Release Planning

In addition to faster deployments, release planning frequency was also increased. A survey by Puppet and DORA showed that organizations with high-performing DevOps enjoy 22% less time spent on unplanned work, 3x lower change failure rates, and enhanced employee engagement [37]. Client Marketing incrementally increased their original release planning cadence of once per quarter to five times per year, then once per month, and then eliminated it with the implementation of continuous planning.

9. Summary

Rocket Mortgage accelerated feature delivery by aligning teams to business-value stream-focused release trains, providing clarity on organizational objectives, and giving teams autonomy to hit those objectives. By also using Scaled Scrum, Client Marketing implemented a MetaScrum to synchronize product backlogs and stakeholders across teams, and the Scaled Daily Scrum to synchronize the leadership team on impediment removal. The team has shown that Scaled Scrum and Scaled Agile frameworks are fully compatible -- and companies can achieve great results by leveraging them together.

With the caveats noted in the Future Research section of this paper, the authors believe the steps taken by Client Marketing as outlined in this paper to be generalizable to other companies and industries for several reasons. First, Client Marketing's success was predicated by using Scaled Agile and Scaled Scrum "by

the book." No customizations were made to either framework to match the processes of Client Marketing. Instead, Client Marketing changed to follow the frameworks. Second, Client Marketing implemented standard software development procedures such as CI/CD [38-41]. And third, Client Marketing uses commercial off-the-shelf (COTS) software, such as CirrCI, Docker, AWS, and Sonarqube.

10. Future Research

The learnings from the Client Marketing Release Train apply to the broader organization. A movement is underway towards wider MetaScrum implementation, Scaled Daily Scrums, delivery focus, and fewer roles.

Two significant trends are relevant to Agile transformations in all industries. First, Rocket Mortgage is moving to business units based on value streams where all teams can directly see their effect on organizational performance. Second, expansion of continuous delivery to enable automated rollout and testing of a subgroup followed by an automatic rollback in the event of problems or automated rollout to a broader base in the case of success.

The effect of value stream organizations with continuous delivery eliminates excessive overhead in release planning and enables the organization to respond more quickly to client requests and market changes.

The large number of intervention components makes it difficult to ascertain which strategies from the black box contributed to which delivery improvements (i.e., measuring the role MetaScrum or SDS played in improving cycle times vs. DevOps practices vs. staffing changes). Future research could be directed at understanding the salience of individual strategies in affecting cycle times. There was also limited availability of pre-Agile transformation data at Rocket Mortgage. Proprietary information also limited the amount of shareable data. Finally, there is limited previous research on combining Scaled Agile and Scaled Scrum frameworks. The authors encourage continued study of the findings and practices outlined in this paper and believe they are a strong foundation for future research.

10. References

- [1] C. Handscomb, D. Mahadevan, L. Schor, M. Siebere, E. Naidoo, and S. Srinivasan. (2020, 21 May). *An Operating Model for the Next Normal: Lessons from Agile Organizations in the Crisis*. Available: <https://www.mckinsey.com/business-functions/organization/our-insights/an-operating-model-for-the-next-normal-lessons-from-agile-organizations-in-the-crisis>
- [2] M. Reeves, K. Whitaker, and T. Deegan, "Fighting the Gravity of Average Performance," *MIT Sloan Management Review*, 2020.
- [3] O. Mikhieieva and K. Stephan, "A Retrospective on Agile Transformations: Survey Results on Agility of German Organisations," in *2020 IEEE European Technology and Engineering Management Summit*, 2020.
- [4] S. Wu. (2021, May 31). *Cerno*. Available: https://www.scaledagile.com/case_study/cerno/
- [5] R. Barnahor. (2021, May 31). *SAFe Case Study: EdgeVerve Systems*. Available: <https://www.scaledagileframework.com/case-study-edgeverve-systems/>
- [6] S. Jagadeesan. (2021, May 31). *Case Study: Royal Philips*. Available: <https://www.scaledagileframework.com/royal-phillips-case-study/>
- [7] L. Christopher and M. Vries, "Selecting a scaled Agile approach for a Fin Tech company," *South African Journal of Industrial Engineering*, vol. 31, pp. 196-208, 2020.
- [8] H. Takeuchi and I. Nonaka, "The New New Product Development Game," *Harvard Business Review*, 1986.
- [9] K. Schwaber and J. Sutherland, "The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game," Scrumguides.org November 2020.
- [10] J. Sutherland and ScrumInc, "The Scrum At Scale® Guide: The Definitive Guide to the Scrum@Scale Framework Version 2.3," Scrum Inc., Cambridge, MA 2021.
- [11] D. Rigby, J. Sutherland, and H. Takeuchi, "Embracing Agile," *Harvard Business Review*, May 2016.
- [12] J. Shook, "How to Change a Culture: Lessons From NUMMI," *MIT Sloan Management Review*, vol. 51, 2010.
- [13] D. K. I. Sobek and A. Smalley, *Understanding A3 Thinking: A Critical Component of Toyota's PDCA Management System*. New York: Productivity Press, 2008.
- [14] J. Shook and Womack, *Managing to Learn: Using the A3 Management Process to Solve Problems, Gain Agreement, Mentor and Lead*. Cambridge MA: Lean Enterprises Institute Inc., 2008.
- [15] E. Elbourne, *Fundamentals of Industrial Administration: An Introduction to Management*. London: MacDonald & Evans, 1949.
- [16] E. Baker, *The Symphony of Profound Knowledge*. Bloomington IN: iUniverse, 2016.
- [17] T. Ohno, *Toyota Production System: Beyond Large-Scale Production*: Productivity Press, 1988.
- [18] M. Tromans, "DEPD-PKT Software Team Process Improvement (A3)," Toyota Kentucky 2016.
- [19] A. Smalley, *4 Types of Problems*. Cambridge MA: Lean Enterprise Institute, 2018.
- [20] ScaledAgile. (2021, June 1). *SAFe DevOps Series*. Available: <https://www.scaledagileframework.com/devops/>
- [21] G. Hermkes and L. Quintela, *Scaling Done Right: How to Achieve Business Agility with Scrum@Scale and Make the Competition Irrelevant*. Berlin: Behendigkeit Publishing, 2020.
- [22] J. O. Coplien and N. Harrison, *Organizational patterns of agile software development*. Upper Saddle River, NJ: Pearson Prentice Hall, 2005.
- [23] J. O. Coplien, "Borland Software Craftsmanship: A New Look at Process, Quality and Productivity," in *5th Annual Borland International Conference*, Orlando, FL, 1994.
- [24] C. Verwijns and D. Russo, "A Theory of Scrum Team Effectiveness," *IEEE Transactions on Software Engineering*, May 26 2021.
- [25] S. Daukus, "Test and fix time inside vs outside a sprint.," Cambridge MA: Scrum Inc, 2016.
- [26] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*: Addison Wesley, 2010.
- [27] ScaledAgile. (2021, June 3). *Case Study: Telstra*. Available: www.scaledagileframework.com/telstra-case-study/
- [28] F. Verbruggen, J. Sutherland, J. M. van der Werf, and S. Brinkkemper, "Process Efficiency – Adapting Flow to the Agile Improvement Effort," in *52nd Hawaii International Conference on System Sciences*, Hawaii, 2019, pp. 6981-6987.
- [29] C. R. Jakobsen and J. Sutherland, "Scrum and CMMI Going from Good to Great," in *Agile Conference, 2009. AGILE '09.*, 2009, pp. 333-337.
- [30] M. Venema. (2021, 29 August). *6 Scaled Agile Frameworks – Which One Is Right For You?* Available: <https://www.digite.com/blog/scaled-agile-frameworks/>
- [31] J. Sutherland, "Future of Scrum: Parallel Pipelining of Sprints in Complex Projects," presented at the AGILE 2005 Conference, Denver, CO, 2005.
- [32] J. Sutherland and J. Coplien, *The Scrum Book: The Spirit of the Game*: Pragmatic Bookshelf, 2019.
- [33] J. Johnson, *Chaos 2020: Beyond Infinity*. Boston MA: Standish Group, 2020.
- [34] H. Kniberg. (2019, June 7). *Spotify: A Scrum@Scale Case Study*. Available: <https://resources.scrumalliance.org/Article/spotify-scrum@scale-case-study>
- [35] M. Fowler. (2010, 20 Sep). *BlueGreenDeployment*. Available: <https://martinfowler.com/bliki/BlueGreenDeployment.html>
- [36] J. Doerr, *Measure What Matters: How Google, Bono, and the Gates Foundation Rock the World with OKRs*: Portfolio, 2018.
- [37] Puppet and DORA, "State of Dev/Ops Report 2016," Puppet 2016.

- [38] Red Hat. (2018, 29 August). *What is CI/CD?* Available: <https://www.redhat.com/en/topics/devops/what-is-ci-cd>
- [39] Cisco. (2021, 29 August). *What is CI/CD?* Available: <https://www.cisco.com/c/en/us/solutions/data-center/data-center-networking/what-is-ci-cd.html>
- [40] A. Crawford. (2019, 29 August). *DevOps*. Available: <https://www.ibm.com/cloud/learn/devops-a-complete-guide>
- [41] Amazon. (2021, 29 August). *What is DevOps?* Available: <https://aws.amazon.com/devops/what-is-devops/>