

# Quality Guild: Creating a Culture

Philip Daye, Jr.

philip\_daye@ultimatesoftware.com

## Abstract

Organizations adopt Agile methods to reap the benefits of shortened feedback loops and a quicker release cadence. Agile teams are made up of cross-functional people formed around a product or feature. This often leads to teams that focus almost exclusively on their individual product areas, which easily become siloed. This is a hidden cost to adopting Agile practices.

Isolated teams will seek solutions for similar challenges, unaware of what others have already done. Fragmented solutions make collaboration difficult and put the health of the product at risk. This becomes increasingly troublesome as the organization grows and tries to scale development. In a culture of silos, how does an organization provide guidance and standards to its teams?

Many organizations attempt to solve this problem by creating a centralized team focused on quality processes, but it too often becomes siloed as well. This new, centralized team often has difficulty penetrating the isolation of Agile teams, and thus, their proposed solutions do not reflect the needs of those they serve. As quality is decentralized, and quality and test engineers are federated out to teams, community leadership becomes an increasingly important mechanism for preserving enterprise level coordination and cooperation.

This paper describes the process one organization, Ultimate Software, undertook, to build a company-wide quality guild, which promotes the shared interests, challenges, and goals of the quality professionals within the individual Agile teams. The guild collaborates to build and evaluate new tools, propose improvements to testing practices, and share findings with the rest of the organization. By breaking down the barriers, the quality guild promotes a sustainable and consistent culture around quality, with less duplication of effort, expanded opportunities for individual engineers, and improved operations.

## Biography

*Philip Daye is a Software Test Lead for Ultimate Software, a developer of HCM software and a perennial on Fortune's Best Companies to Work For list. Philip has more than two decades of experience in software development, testing, and delivery for companies of all sizes. From his earliest days as a software tester to today, Philip has pursued a passion for quality, which has led him to hone his craft by studying and researching the latest advances in the field, applying it to his current work, and then sharing it with others.*

*In his current role, Philip helps document quality practices, consults with teams on how to apply those practices, and develops and delivers training on testing and test techniques. As an active participant in the broader testing community, he has taken on co-organizing the South Florida Test Automation meetups.*

Copyright Philip Daye, Jr. 2019

# 1 Introduction

The Merriam-Webster Dictionary defines culture as “the set of shared attitudes, values, goals, and practices that characterizes an institution or organization” (Merriam-Webster Dictionary, n.d.). This culture is “the integrated pattern of human knowledge, belief, and behavior that depends upon the capacity for learning and transmitting knowledge to succeeding generations” (Merriam-Webster Dictionary, n.d.). For a company, its “internal culture may be articulated in its mission statement or vision statement” (WhatIs.com, n.d.).

72% of CEOs that participated in an Inc. survey said that “Maintaining company culture” was “the hardest part about managing a growing workforce” (Inc., n.d.). As a company grows, subcultures will emerge around shared work areas, common job types, or mutual project boundaries (such as a feature-based development team) (Leading Agile, 2015). This paper presents an approach for maintaining cross-team, functional collaboration within a projectized organization. The supporting case study specifically examines how a common quality subculture was intentionally maintained in an otherwise decentralized Agile environment with fully distributed quality assurance personnel and responsibilities. How can the shared values of a subculture around roles be aligned with those of the teams and the overarching goals of the company?

## 2 Ultimate Software

Ultimate Software is a Human Capital Management (HCM) solution provider. The company’s “People First” principle permeates all aspects of the business, including our approach to our employees and customers: “At Ultimate Software, we take care of our employees, and we know it will show in the way they take care of you” (Ultimate Software, n.d.). Our corporate mission statement continues the theme of improving “the personal work experience for you and your people - the power behind the business” (Ultimate Software, n.d.). To be People First, we recognize that our organizational structure must be responsive to our customers and to our employees. As such, there must be a balance between the needs for product development and people development. Career development is often built upon improving skills within a functional domain such as software development or quality engineering. Being People First requires that the organization provides clear paths for employees to improve and expand their functional skills and responsibilities.

The development organization has adopted Agile, with Kanban as the work-in-progress management tool. Accordingly, teams include personnel to specify, architect, produce, and validate product features. These teams “produce a working, tested increment of product” (Leading Agile, 2015). As HCM encompasses a complex domain, several functional teams are often grouped together to work on a common product sub-domain (e.g., Pay, Tax, or Talent).

At one point, all development teams were located at Ultimate Software’s corporate headquarters in Weston, FL. Maintaining a quality culture was a simple process as quality engineers would see each other in the hallways and open areas. Inter-team coordination along functional lines was as easy as gathering in a single conference room. Quality issues were addressed quickly.

As the company began to grow rapidly, the development teams could no longer be located in the same building. Where culture initially grew organically from daily interaction, a more deliberate approach was needed to promote and share that original quality culture. This was addressed through the use of a centralized command structure where a few managers and architects dictated monolithic solutions. While this maintained a relatively common solution set for a short period of time, as the company continued to scale, such a structure quickly became intractable as product lines diverged. Addressing these new product needs, teams adopted new technologies and architectures, which required greater team autonomy.

Through growth and acquisition, development has spread to both coasts of the United States as well as into Canada and Europe. Each new office location adds greater physical separation between teams. Where it once was possible to bring all of the testers in the organization together in one conference room to share information and hold discussions, the logistics of space and time no longer make this possible. As teams grow, it becomes increasingly difficult to have conversations together and ensure that every voice is heard. Where the quality organization once had a unified culture, it has split into subcultures, overshadowed by the individual teams' cultures.

As Ultimate Software's offerings have grown to meet the increasing demands of the HCM market, the need for inter-team coordination has also grown. It's no longer possible for any single person to be capable of comprehending all of the sub-domains and architecture used across the company's products. The company is no longer able to rely on architects or centers of excellence to satisfy all coordination requirements.

The core values that define a culture depend on them being taught and shared. As test engineers were assigned to feature teams, and as new people joined, the sharing became fragmented. With less cross-team interaction, teams began to pursue solutions for similar problems without understanding what other approaches were available, or that their current pursuit had been tried. As an example, a commercial test tool for a legacy product was no longer available or supported. It would not be usable on newer operating systems. Either a replacement would be needed, or regression testing would become a manual effort - adding a number of days to a release. Several teams worked independently towards a solution. In itself, this is not bad, but some of the approaches would not have worked for all teams. Also, teams ended up repeating work. This is the impact of siloed teams - no sharing of information, duplication of effort, and solutions that don't meet all teams' requirements.

What if the same fragmentation was occurring in quality practices and principles? What if quality was being defined outside of the community of testers, and without consulting the test architects that had crafted a set of standard practices best suited for the needs of the organization?

A solution was needed that would allow test engineers to share information, tools, and practices and through which organizational standards could be communicated out to the teams.

## **3 The Quality Guild**

### **3.1 Community of Practice**

Agile teams are cross-functional to facilitate the delivery of increments of the product, but there's value in test engineers communicating with their peers, sharing their multiple experiences and knowledge. The Community of Practice is one way to support function-centric discussions and collaboration (Scaled Agile, Inc., 2018).

Communities of Practice are "organized groups of people who have a common interest in a specific technical or business domain. They collaborate regularly to share information, improve their skills, and actively work on advancing the general knowledge of the domain" (Scaled Agile, Inc., 2018).

### **3.2 Scaling Agile: Spotify**

One example of Communities of Practice is found in the organizational model used by Spotify and laid out in "Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds" (Crisp's Blog, 2012).

Spotify organizes feature teams into Squads. Squads working in related areas are organized into Tribes (Crisp's Blog, 2012). These Squads and Tribes are analogous to Ultimate Software's Teams and Domains. The difference is that Spotify then has Chapters that organize along a function within a Tribe. The Chapter, combined with the Squad and Tribe, form a management matrix with the individual contributor reporting to their Chapter lead (Crisp's Blog, 2012). Ultimate Software does not have this matrix structure.

Finally, the concept of the Guild is to bring together those with a common interest from across Squads and Tribes - from across the organization. Spotify defines the Guild as “a group of people that want to share knowledge, tools, code, and practices” (Crisp’s Blog, 2012). This is the Community of Practice. The Squads and Tribes maintain their autonomy, while the Guild allows for sharing and learning that cuts across both. A culture is built on the sharing and teaching of common values and goals.

### 3.3 Ultimate Software: The Quality Guild

Figure 1, below, is a diagram of the Guild structure at Ultimate Software. Test engineers from feature teams, across Domains, are able to gather and discuss issues related to quality practices and tooling.

As noted earlier, centralized control doesn’t scale and often becomes siloed. Guilds are decentralized and self-organizing. In the book *The Starfish and the Spider*, the authors describe two persons critical to launching decentralized groups: the catalyst and the champion. The catalyst is the person who gets a decentralized organization launched and then cedes control. The champion pursues the idea and takes it to the next level (Brafman and Beckstrom 2006).

The Quality Guild’s catalyst was Tariq King, Distinguished Architect and Sr. Director of Quality Engineering. He provided the vision and the critical management support needed to launch the guild. The Quality Guild also required a champion, someone who could act as “guild coordinator” and who would organize and facilitate regular meetings. The author was selected for this role.

Initially it was decided to invite only one test engineer from each feature team, preferably a test lead or a test engineer acting in that capacity. This was done to keep the size of the group manageable as it got started.

The initial meetings were run similar to a “meetup,” with a fixed agenda based around a presentation. While this was successful at first, it quickly became difficult to arrange presentations and the topics for discussion began to stagnate. There was also a sense that too many of the discussions focused around one tech stack and some members felt they weren’t getting any true value from the meetings. This led to a fall in attendance.

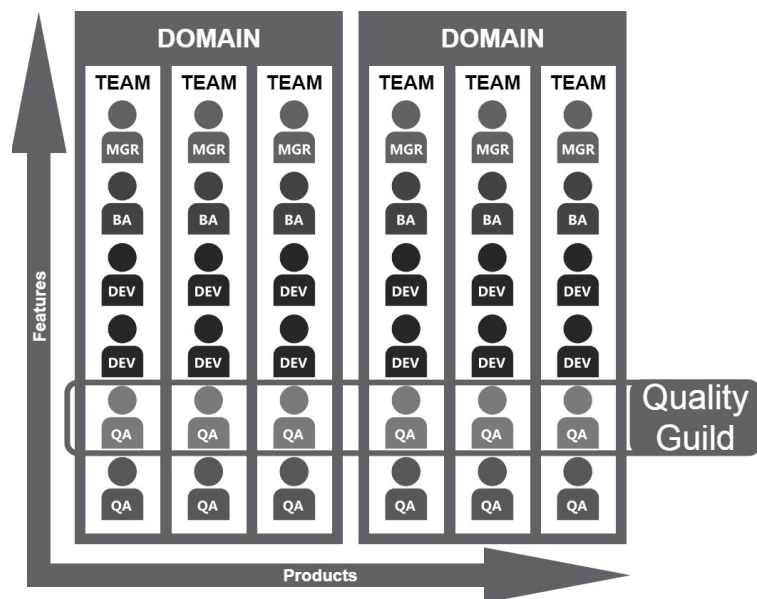


Figure 1: Teams, Domains, and the Quality Guild

In addition to the meetings, a channel was setup in Slack to create a more immediate way to share information. This was a place to not only ask questions and share ideas, but to share links to interesting posts and tools. With Slack being a primary communication tool at Ultimate Software, it became the natural choice for the guild.

Three approaches were taken to address the problems with the meetings:

First, membership was opened more broadly. While it still isn’t open to every test engineer, those who express an interest are extended an invitation.

Second, instead of soliciting presentations and using a static agenda prepared by the facilitator, we adopted Lean Coffee to generate our agenda. Lean Coffee is referred to as an “agenda-less meeting” (Lean Coffee, n.d.) because the agenda is not fixed in advance. Typically, participants would write topics on cards and then the cards would be voted on and discussions would start with the topic that got the

most votes. We chose to use an electronic board that stays up between meetings, allowing members to add topics whenever they want, and also to vote on what interests them. Some variations of Lean Coffee will timebox discussions, allowing for votes for more time. We decided not to restrict the discussions that way. Since we use an electronic board, we also carry topics over from meeting to meeting - along with their collected votes.

Third, the Quality Guild has started forming Working Groups, chartering them to address particular issues. They are expected to complete the work and disband. The length of time a Working Group will exist is based on the work to be done. Participation is voluntary, so those involved are generally more passionate about the topic. Each Working Group chooses its own chairperson to organize and facilitate the work. At the time of this paper, there are four active Working Groups:

- **Test Case Management:** Gathers requirements for tooling to support test case creation and maintenance, reviews available options, and proposes solutions.
- **Root-Cause Analysis:** Develops standards and processes for collecting root-cause analysis data on escaped defects and develops tools and reporting to use in process improvement.
- **Test Coverage Metrics:** Develops the process to collect and report data regarding teams' efforts around testing of features.
- **Accessibility Testing:** Develops architecture and quality guidance addressing accessibility testing, architecting for accessibility, and accessibility conformance reporting.

Working Groups are expected to give regular updates to the Quality Guild on their progress, which also gives an opportunity for the broader community to ask questions and provide feedback on the work.

"Typical guilds live and die by the participation and motivation of their members" (AZ Central, n.d.). Fortunately, these initiatives have improved participation and the Quality Guild is growing.

## 4 Results

The Quality Guild, on its own or working in conjunction with Architecture, has provided key insights, assistance, and work on documentation, test case management, root-cause analysis, test coverage metrics, accessibility testing, and training.

### 4.1 Documentation

Ultimate Software is developing an internal site for documentation related to our architecture, development, and quality standards.

The Quality Guild provided source material on each team's internal processes for reviewing automated GUI-level system tests. This formed the basis for a document on standards for these tests. Where there was consensus, the standard was added to the document. Those where there might be disagreement, and areas where there was no guidance, were reviewed by a quality architect for input. The final document was reviewed by the Quality Guild. The document is now live and is considered the standard for Ultimate Software.

In addition to this direct contribution, guild members have also participated in other quality standard documentation being written by the quality architecture team. Their assistance has included identifying the priority of documents to be created and reviewing the documents before internal publication. The documents developed and reviewed have included guidance on test case authoring, the test pyramid, exploratory testing, risk-based testing, test planning, performance testing, and on production readiness from a quality perspective.

Some of the documentation codifies existing practices and will serve as training material for new testers (and a reference source for everyone). Others, such as risk-based testing, are practiced by some teams, but adoption will increase with published guidance. Since teams use different technologies and architectures, many of the documents have to be written for a broad audience. For instance, the test

pyramid material does not define a single pyramid for all teams, but instead discusses the rationale behind the pyramid and provides guidance on how to use it as a heuristic for developing an effective test portfolio.

The documentation site is still relatively new, so the Quality Guild members reviewing documentation are helping to drive adoption of the site. Likewise, quality architecture is actively encouraged continued involvement, including contributing documents and assisting with future maintenance.

## **4.2 Working Groups**

### **4.2.1 Test Case Management**

A prior informal investigation of this topic, held before the working group existed, had generated a set of requirements that must be met by any new tool acquired or built for the authoring, maintenance, execution, and reporting of manual test cases. These requirements were reviewed and refined by the working group and included the following:

- integration with tools, such as our issue tracking system (Jira) or an API to provide the integration
- ability to version test cases
- a configurable interface
- report generation
- ability to execute and collect results for manual tests

Based on the requirements, a select number of vendors were contacted about trials of their tools. These tools were deployed, and the working group reviewed and scored them against the requirements. The tools showed some level of integration with Jira, but little else in the way of APIs for extensibility. Most had interfaces that were not intuitive and often cumbersome to navigate and use.

We also surveyed open source test management tools, without finding a single tool that met a sufficient number of our requirements to justify selecting it for further evaluation.

Due to the somewhat lackluster response to all of the commercial tools reviewed, and the lack of open source alternatives, the chairperson, working with another engineer, developed a prototype of an internal tool, as an additional option. This prototype is currently being reviewed by working group members. The members of the full Quality Guild will be invited to comment on the prototype as well.

The working group is on track to recommend either funding the acquisition of a commercial tool, or the development of an internal tool within the next month. A problem that had been discussed for a few years will have been moved forward by the working group within a matter of months.

### **4.2.2 Root-Cause Analysis**

The purpose of root-cause analysis of defects is to determine what happened, why it happened, and how to reduce the likelihood of another defect like it happening again in the future. Defects that escape the development process and are discovered in a production environment are the costliest to fix in terms of time taken from new features and damage to the company's reputation with its customers.

This working group quickly developed guidance based on IBM's Orthogonal Defect Classification (Chillarege, et.al. 1992) for collecting relevant information about escaped defects. The guidance used questions like: What happened? Why did the defect occur? When should it have been found? and How can it best be avoided in the future? Using this guidance and model, a pilot program was created, engaging seven development teams. The pilot data was collected in Google Forms.

With the pilot completed, a retrospective was held and some of the data collected was reviewed. One key point was that the collection of the data caused the teams to have discussions around their defects and how they might prevent similar ones in the future. Since this is a key goal of root-cause analysis, everyone agreed this was a positive result. The teams did note that collecting the data in a Google Form

instead of in the defect did not fit their workflow. None of the teams showed a desire to continue to collect this data after the pilot ended, unless the data could be collected in the centralized issue tracking system used by Ultimate Software (Jira).

The working group now has two primary tasks to complete:

- First, it must work with the Jira administrator to add fields to the defect form so that root-cause data is collected along with the defect itself, as the teams clearly desire.
- Second, they must develop guidance on how teams must collect and analyze the data and develop experiments on improving the team's development processes to prevent particular classes of bugs in the future.

#### 4.2.3 Test Coverage Metrics

This working group is developing a model for collecting data indicating how well a feature has been tested. The first task they had to accomplish was to define, for the sake of knowing what to measure, what a feature is: a business facing capability, or an internal process. Building on this, they have created a draft proposal for guidance on collecting data to build a coverage metric. It was decided that feature coverage would be somewhat aligned with the concept of the test pyramid.

Metrics to collect include:

- Unit level test coverage
- Integration level test coverage
- System level test coverage
- Compliance test coverage
- Performance test coverage

**Note:** Compliance testing includes non-functional concerns other than performance, such as security, data privacy, accessibility, and regulatory.

For unit level test coverage, teams will use code coverage reporting from SonarQube to provide the coverage percentage.

Code coverage is a quantitative metric. There has been no consensus on similar quantitative metrics for the other levels of testing to be reported, so teams will be asked to provide a qualitative metric for each of those. The data is to be provided in the form of a percentage of coverage they believe their team has achieved for a given feature.

Given the complexity of determining this metric, the working group is currently focusing on two intermediate steps: ensuring all teams are reporting unit test code coverage via SonarQube and developing tooling that will assist in the collection of qualitative metrics. Part of that tooling is a Taxonomy Manager that allows teams to build a tree-like structure of their product, grouping related features together in branches and leaves. There is also consideration of adding risk analysis data to this taxonomy so that test coverage of individual features are weighted by risk to provide further insight into the overall health, based on test coverage of a product.

#### 4.2.4 Accessibility Testing

The World Wide Web Consortium (W3C) has developed the Web Content Accessibility Guidelines (WCAG) "with a goal of providing a shared standard for Web content accessibility" (W3C, n.d.). Most countries that have adopted accessibility standards base them on adherence to WCAG.

The company cannot meet either its People First commitment, or its legal obligations, if our solutions are not broadly accessible. This working group is focused on developing architecture and quality guidance addressing accessibility testing, architecting for accessibility, and accessibility conformance reporting.

Members of this working group have delivered an introductory class on accessibility to a team designing interface controls for future projects. The goal is to have these controls support accessibility “out of the box.” This introductory material will be used as the foundation to develop more in-depth training for developers and testers throughout the organization.

The working group is currently preparing a proposal for an approach that will aid teams in determining their current level of conformance, planning and executing any remediation required, and then remaining in conformance.

Work was completed to integrate an open source accessibility test library, aXe (<https://github.com/dequelabs/axe-core>), into Aeon (<https://github.com/UltimateSoftware/aeon>), a test framework developed in-house, so that teams could begin to include automated accessibility checks into their system tests that run against the web interface of UltiPro, Ultimate Software’s core product.

Another project that is currently being pursued is Agent A11y. Agent is an open source AI-driven testing agent that explores web pages (<https://github.com/UltimateSoftware/AGENT>). Leveraging the Agent platform to run a static analysis tool such as aXe will allow us to autonomously collect information on accessibility issues. Additionally, Agent A11y is able to perform certain dynamic accessibility tests such as tab order and element focus checks.

### **4.3 Training Classes**

One of the ways the Quality Guild perpetuates the quality culture at Ultimate Software is to share knowledge. That includes how we train our own people on testing.

Internal training is an integral part of the career experience of employees of Ultimate Software in general, and quality engineers in particular. Every tester joining the company is invited to attend a black box testing techniques class. Additionally, a class on white box testing techniques is available. Members of the Quality Guild have been instrumental in reviewing and piloting a new class on Session-Based Test Management. The feedback was extremely valuable and is being used to complete the final draft before rolling it out to all of our test engineers.

Originally, all test-related training was done by test architects, but over time it became clear that more trainers would be needed. Test leads have been selected to be prepared to deliver the training themselves. This is an important part of their career growth, opening new opportunities as their impact on the organization increases. This is similar to how the Marine Corps “tap outstanding performers to fill instructor slots, regarded as one of the high-profile tours of duty” (Freedman 2000). Because they will be responsible for teaching newer testers these techniques, leads are challenged to deepen their own knowledge of testing and quality engineering.

## **5 Future Work**

The Quality Guild is like exploratory testing in that we try an experiment and see what happens. Based on the results of that experiment, we may keep doing something, or we may move on and try something else. Looking ahead, areas that we will be experimenting in are Attendance and Engagement, Diversity, and Working Groups.

### **5.1 Attendance and Engagement**

Participation in the Quality Guild is strictly voluntary and in addition to a test lead’s usual work, therefore we cannot penalize someone when they don’t attend meetings. That said, we have tracked attendance for almost a full year now. The purpose is not to be concerned with individuals, but to see which teams and domains aren’t being represented. Those teams will miss critical information that’s shared and have no input into decisions that are made if they don’t have a presence at the meeting.



We also want to continually review attendance to determine if there are people on the invite who have been promoted or moved to a different functional role and are no longer interested in participating. Replacing them allows us to focus on improving representation from the teams.

## 5.2 Diversity

As of August 2016, Ultimate Software reported that 49% of the total workforce were women (Ultimate Software, n.d.). Unfortunately, our initial invite list did not reflect this diversity. We noted it at the time but were unable to identify additional qualified individuals already in the company. Since then, opening the membership to include those expressing an interest has improved the situation, but there's a long way to go. Opening it further, as planned, may also improve the diversity of the guild.

Beyond passive approaches, one of the purposes of the Quality Guild is to provide mentoring to engineers who want to grow and could eventually become leaders. Greater outreach needs to be fostered and diversity needs to be an issue kept before us.

## 5.3 Working Groups

Working Groups would benefit from a greater emphasis upfront on their charter, including stipulating completion criteria. The current working groups had to take a vague requirement and refine it into a mission with a clear goal, which delayed their addressing the problem they had gathered to solve. Also, an early, clear charter might attract other members to participate.

Additionally, all of the current chairpersons come from central teams, such as Architecture and the Quality Center of Excellence. There is a goal to make sure that as often as possible, the chairperson, or a co-chairperson, come from a feature team, since they are the most impacted by the guidance and tools produced. This is also an ideal place to mentor more junior members.

## 6 Conclusion

This paper has examined Ultimate Software's Quality Guild as a way to promote a shared culture among quality professionals from across the feature teams that build and deliver software. That software improves "the personal work experience" for our customers and their employees - "the power behind the business" (Ultimate Software, n.d.).

We looked at Spotify's model for communities of practice, which they named guilds, as a way to create a decentralized team that could sustain their culture while addressing the issues that impact the feature teams. In this way, teams would learn from each other.

Working together, the Quality Guild has helped with the development of documentation to provide guidance on test approaches that work across Ultimate Software. Working groups have been created to address specific issues that teams are facing (the need for test case management tooling and root-cause analysis to drive process improvement) to organizational issues (such as the need for measures of product health and assessing the accessibility of our products, a moral and legal requirement). The guild is also concerned with the development of the individual quality engineer and so it is actively involved in the development of classes and preparing test leads to teach, as well as to mentor others.

As a result of discussions within the guild, new tools have been created, or are in the process of being created, that will solve problems teams are currently facing. This includes tooling around test case management, metrics collection, test planning, and accessibility testing.

Culture is about shared values and goals. Ultimate Software has a strong culture built around the principle of "People First." The Quality Guild is an outgrowth of that culture, through which we have instilled and grown a sub-culture of quality that also strive to achieve this principle.

## References

- Agile Coffee. n.d. "What is Lean Coffee?" Accessed June 8, 2019. <http://agilecoffee.com/leancoffee/>.
- Agile Transformation. 2015. "What is an Agile Team and How Do You Form Them?" Accessed June 8, 2019. <https://www.leadingagile.com/2015/02/what-is-an-agile-team-and-how-do-you-form-them/>.
- AZ Central. n.d. "Culture and Subculture in Business." Accessed June 15, 2019. <https://yourbusiness.azcentral.com/culture-subculture-business-3801.html>.
- Brafman, Ori and Rod Beckstrom. 2006. *The Starfish and the Spider: The Unstoppable Power of Leaderless Organizations*. New York, NY: Penguin Group.
- Chillarege, Ram, I.S.Bhandari, Jarir Chaar, M.J.Halliday, D.S.Moebus, Bonnie Ray, M.-Y.Wong. 1992. "Orthogonal Defect Classification - A Concept for In-Process Measurements." *IEEE Transactions on Software Engineering* 18: 943 - 956.
- Crisp's Blog. 2012. "Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds." Accessed June 8, 2019. <https://blog.crisp.se/2012/11/14/henrikkniberg/scaling-agile-at-spotify>.
- Freedman, David H. 2000. *Corps Business: The 30 Management Principles of the U.S. Marines*. New York, NY: HarperBusiness.
- Hootsuite. 2014. "Guilds: Get Stuff Done Together." Accessed June 8, 2019. <https://medium.com/hootsuite-engineering/guilds-get-stuff-done-together-3d0826209390>.
- Inc. n.d. "Preserve Your Company's Culture As You Grow: 5 Steps." Accessed June 15, 2019. <https://www.inc.com/paul-spiegelman/company-culture-manage-growing-workforce.html>.
- Lean Coffee. n.d. "Lean Coffee Lives Here." Accessed June 8, 2019. <http://leancoffee.org/>.
- LinkedIn Pulse. 2015. "Guilds, Tribes, Chapters & Squads... and SCRUM. What?!" Accessed June 8, 2019. <https://www.linkedin.com/pulse/guilds-tribes-chapters-squads-scrum-what-arjan-bos/>.
- Merriam-Webster Dictionary. n.d. "Culture." Accessed June 14, 2019. <https://www.merriam-webster.com/dictionary/culture>.
- Pink, Daniel H. 2009. *Drive: The Surprising Truth About What Motivates Us*. New York, NY: Riverhead Books.
- Poppendieck, Mary and Tom Poppendieck. 2003. *Lean Software Development: An Agile Toolkit*. Boston, MA: Addison-Wesley Professional.
- Prowareness. 2016. "How to get Great Scrum Guilds." Accessed June 8, 2019. <https://www.scrum.nl/blog/get-great-scrum-guilds/>.
- Scaled Agile, Inc. 2018. "Communities of Practice." Last modified September 21, 2018. <https://www.scaledagileframework.com/communities-of-practice/>.
- Ultimate Software. n.d. "Our Culture." Accessed June 14, 2019. <https://www.ultimatesoftware.com/Ultimate-Company-Culture>.
- Web.com. 2017. "A New Agile Guild Model." Accessed June 8, 2019. <https://medium.com/webcom-engineering-and-product/agile-guilds-the-yodle-way-47dc00f6cd3a>
- WhatIs.com. n.d. "corporate culture." Accessed June 14, 2019. <https://whatis.techtarget.com/definition/corporate-culture>.

World Wide Web Consortium. n.d. "Web Content Accessibility Guidelines (WCAG) 2.1." Accessed July 20, 2019. <https://www.w3.org/TR/WCAG21/>.