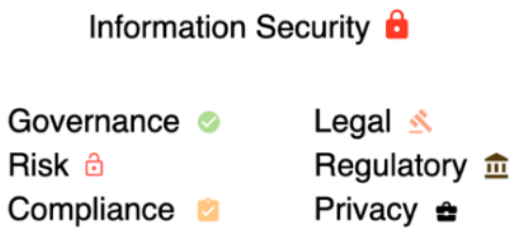
Kevin Kennedy
Brown EMCS 2020

# The Agile Blue Team: Leveraging DevOps Concepts for Continuous Cyber Resiliency

Cybersecurity as a standalone discipline is still in its infancy. The industry as a whole is expected to reach a market cap of 250 billion dollars by 2023 - nearly doubling the current market size[1]. This growth is fueled by a rise in cyber-crime, an expanded regulatory landscape, and more generally the need to protect a plethora of new technologies such as cloud computing, IOT, mobile, autonomous vehicles, the rise of smart homes and more. The continued proliferation of technology footprint will require unique protections from cyber-attacks as well as an interdisciplinary approach to cyber program management within organizations.

For highly regulated industries such as banking or financial services, there are well established information security norms of governance, risk, compliance, legal, regulatory, and privacy. The Information Security function utilizes standard risk frameworks such as NIST or ISO to drive top down program management. The Information Security function typically writes policy, and generally oversees the governance of cybersecurity program. In some cases, there will even be an audit department sitting above the information security function and the C-Suite and Board of Directors even further above that. These norms are, therefore, well established and structurally defined at the organizational level.

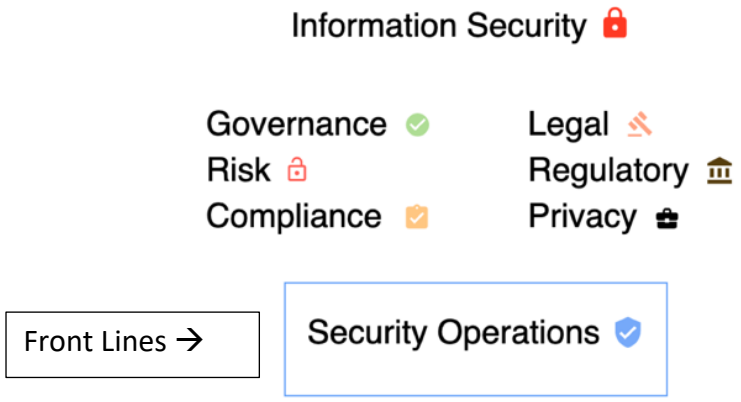
Typical Information Security Program Structure




However, what about the lowest level of security operations? How does a structurally sound information security organization communicate the program directives down to engineering level? This lack of integration and communication from the organization level to the more tactical / technical level is the primary disconnect in most information security programs today. Specifically, this paper will attempt to define better ways of working at the lowest level of operations to ensure that the tactical and technical work is in lockstep with the overall cyber and risk management program. In particular, this paper seeks to define some methodologies that security operations engineers can leverage to ensure they are working on the most impactful

[1] Statista. "Global Cybersecurity Market Forecast 2017-2023." Accessed January 20, 2020. https://www.statista.com/statistics/595182/worldwide-security-as-a-service-market-size/.

work to drive down organization risk. These methodologies will draw strong parallels between the software engineering discipline and the methodologies utilized by software developers.




Looking back in recent history we can see parallels between the software development discipline and cybersecurity as we know it today. Software development as a standalone field exploded as a result of the dot com boom in the late 90's and early 2000's. Propelled by the need to build applications faster, with less effort, and greater precision, software developers adopted a standardized methodology ubiquitously called "The Agile Methodology." This methodology, at its core, was a new way of thinking about "how" to write better software. It bore various processes and frameworks which ultimately helped the software development community to write software, faster, more consistently, more collaboratively, and with a much greater focus on delivering direct value to the end user. Agile methodology was an extreme focus on the "how" of software development. Agile transcended languages, frameworks, and technologies. Instead, it described a set of core principles about how the "work" should actually get done.

The cybersecurity field, in contrast, has no such "methodology" on the "how." How should cyber defenders work when they to get in the trenches trying to protect organizations? Organization level frameworks such as NIST and ISO help organizations to develop a deep and comprehensive cyber program, however, they do little for the blue team defender trying to protect an organization's assets and drive down risk on a daily basis.

Other frameworks do come closer to the tactical. Frameworks such as The Center for Internet Security (CIS) creates a standard for hardening systems for cyber-attacks. CVSS was developed as a standard formal way for communicating vulnerabilities across platforms. Standards such as PCI were developed to help standardize the controls around services that conduct payment card processing. OWASP top 10 exists to define the most common attacks seen in web applications… the list goes on.

Despite the existence of various frameworks noted above, there still lacks a comprehensive "methodology" for the "how" of running a tactical cyber defense program. Existing frameworks are generally prescriptive in nature. They are born out of a response to governance and compliance requirements rather than helping to define a methodology to

tactically defend your organization's assets from cyber-attacks. These frameworks often attempt to identify, define, and protect against your risk and/or measure your organizations security posture relative to its risk profile. They ignore the fact that cyber warfare is really won in the trenches.

Agile development, in contrast, is specifically focused on the about the "how." Agile defines a culture. It better defines how teams should work. How can we write better code? How can we produce more meaningful features? How can we deploy faster and with less bugs? How can we make our applications more resilient and reliable? Agile development is not about governance. Instead, it dictates a cultural shift in the way software developers view the people, process, and technology stack.

There is no equivalent of this in the Cybersecurity field. In part, this is because the cybersecurity discipline is still nascent, and the key roles required for organizations to maintain an effective cybersecurity posture are still being defined and formulated. As the discipline continues to mature, and the market for cyber professionals ultimately gets filled, we will need a comprehensive methodology to bring these teams together in a structured, formal way. Teams will need a methodology about the "how." It seems clear then, that cyber blue teams might benefit from leveraging many of the same concepts as Agile software development teams adopted in response to the dot com boom.

## The Golden Age of Application Development: The Agile Methodology

In the early days of software engineering there were simply languages and platforms. The combination of which provided the tooling that developers needed to build software applications. Software development, therefore, was a skill, and not quite a standalone discipline. In the early 2000's, the industry began to formalize a standard methodology for software engineering which focused less on the tooling, and more about the people and the process of building applications. Thus, "The Agile Manifesto[2]" was born.

In 2001, a group of 17 technologists came together to create this credo. This group defined 4 key principles that all software development teams should subscribe to in order to build better applications:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

[2] "Manifesto for Agile Software Development." Accessed January 18, 2020. http://agilemanifesto.org/.

Agile development, therefore, stresses the importance of the "people" and the "process" behind the software development lifecycle. These are primary considerations over and above the underlying technology platform or programming language used. This cultural shift marked the beginning of a golden age of web application development which expanded the footprint of the internet exponentially over the past 20 years[3]

## People, Process, and Technology Agility: The "How" of Software Development

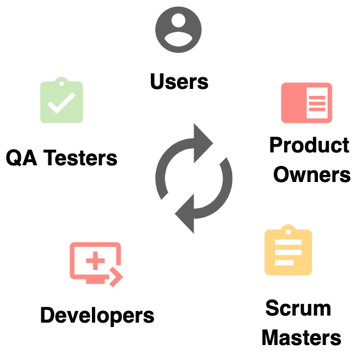
While Agile development became a high-level credo by which developers could now subscribe, it lacked a formal framework in which development teams could actually define in their day to day operations. This led to the creation of a few major frameworks that defined the "how" of the Agile credo. The most widely used today is something called the "**Scrum**" methodology. Scrum (or Kanban) ultimately is the tactical embodiment of the Agile Manifesto.

Scrum, for example, breaks down the software development lifecycle into something called fixed "**sprints**." Sprints are defined periods of development (usually 2-week cycles) where a small amount of use cases or functions are defined, built, tested, and deployed into the product. Before every sprint, a product owner works closely with end users to determine the most impactful changes that would most benefit the application. The product owner then turns these items into "user stories" which typically take the form of a "change," "bugfix", or "new feature." The product owner then takes these "user stories" and works with the development team to enrich the stories with more technical implementation details. The key here is prioritization. Items that are not the most critical or impactful to the business are generally disregarded in favor of the highest priority issues. This creates agility.

Once these issues are finalized and enriched with technical implementation details, the "Scrum Master" (a role akin to a project manager) facilitates the fixed "Sprint." During this timeframe, the "user stories" are developed, tested and deployed into the application. Feedback is then captured from end users in real time. This process is underpinned by a deep collaboration effort amongst all of the key stakeholders. The whole process becomes very democratic. Everyone from key stakeholders, to end users, to developers, to QA testers, to scrum masters all contribute feedback into the process.

[3] "Internet Growth Statistics 1995 to 2019 - the Global Village Online." Accessed January 17, 2020. https://www.internetworldstats.com/emarketing.htm.

Agile Methodology: People



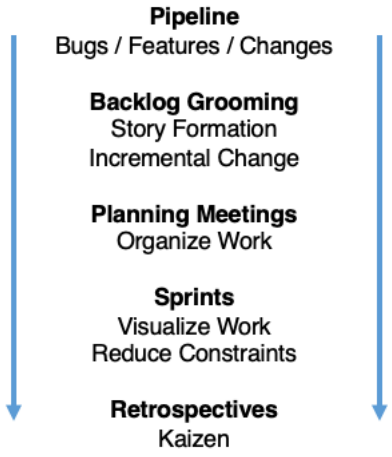

The increasing focus on "people" collaboration gave rise to concepts like "morning standups." This software team staple requires a quick 10 minute "stand up" meeting each day where developers, product owners, and scrum masters have an opportunity to relay any important updates to the rest of the team. Also, it is an opportunity to identify any major roadblocks with the sprint and generally meant so that each developer was aware of what the rest of the team was working on. Following standups, developers often utilized "pair" programming techniques in order to keep cross trained and ensure that the team dynamics were upheld through the development lifecycle.

In order to further emphasize collaboration, Agile teams generally employ a few structured meetings throughout the sprint, or the lifecycle of an entire applications. Firstly, there is an initial "planning meeting" with the entire development group. This planning meeting takes place at the start of each sprint. It is an opportunity for the product owner, or user base, to describe the tickets to the development team. It is where the tactical implementation of each user story is defined and placed in the development queue as part for the next sprint. Issues are broken down into small enough tickets where they can be separately prioritized and assigned to the right developer.

Following the completion of a fixed sprint, the team will come together for a formal structured meeting called a "retrospective." The retrospective will bring together the entire development team, sometimes in consultation with representatives of the user base, and review the work product that was completed throughout the sprint. It is a structured opportunity for anyone on the development team to define any lessons-learned or communicate opportunities to deploy better architectural decisions as part of the next sprint.

Agile scrum teams will also conduct frequent meetings called "**Kaizens**" which is Japanese for "Change for the Good." These meetings do not focus on technology. Instead, they focus on the "process." Kaizens attempt to find areas in the development lifecycle, team dynamics, communications structures which would enhance "how" the team works. For example, a Kaizen might raise awareness that the Sprints are too short in duration (or that the workload going into the sprints are too meaty). A Kaizen might raise awareness that some members of the team are writing code that doesn't confirm to a previously agreed coding standard. Out of the Kaizen comes change. Good change.

Agile Methodology: Process




These Kaizen's were a critical component toward developing tools to help teams automate the software development lifecycle. This led to better technology around the build / test / deploy process and gave rise to concepts such as "Continuous Deployment" and "Continuous Integration." Git repositories were created to help simplify code repositories and IDE's were developed to help developers write better code. Tools like Travis and Jenkins automated the build / deployment, and testing processes. Microservices like Docker further created efficiencies for developers to work independently of IT groups.

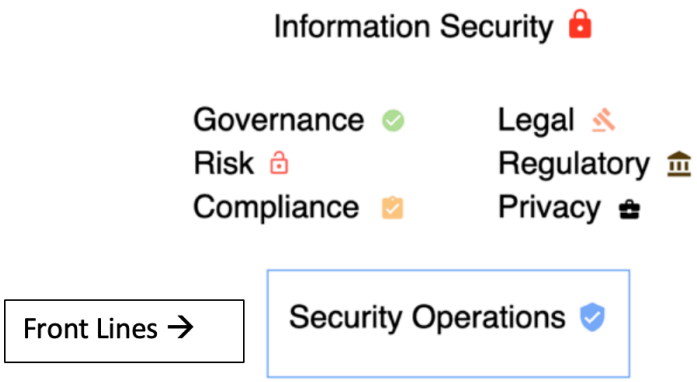
Ultimately, DevOps as we know it today emerged as the standard way to develop applications and run IT operations. It all really started with the Agile credo. DevOps took IT operations and handed it to developers. Developers found ways to continually improve the processes of IT operations through better processes, better tooling, and ultimately a mindset of continuous improvement and efficiency. It is these core outcomes that helped define this new golden age of application programming.

Agile Methodology: Technology

## The Agile Blue Team

The brilliance of the Agile methodology is that it focuses on the "how" of software development. The cybersecurity field, by contrast, has no such methodology on the "how" from a people, process, and technology standpoint. Instead, any standardization in the cybersecurity field is typically a top down approach. At the highest level of the information security there are frameworks such as NIST and ISO to drive top down governance, risk, and compliance strategy. However, little information is provided on the tactical where the security operations group functions. Why? It is the security operations group that is ultimately tasked with the tactical work of protecting an organization from cyber threats.

This gap in the "how" to run a cyber program is no less apparent than in the banking sector. In banking and in other larger, more regulated organizations, the information security program is typically defined in terms of the "3 lines of defense" model. This lexicon is used to articulate how the organization structures it's "cyber defense program." The first line refers to the application owners themselves that are responsible for the risk posture and controls of the applications within the organization. The second line refers to the entity level controls at the information security program level (office of the CISO e.g.). Finally, the third line refers to the audit function to oversee the program defined by the second line. This model, however, does nothing to dictate how the "front line" of cyber defense (security operations) should function.




Despite an abundance of governance, risk, compliance, and frameworks, many security operations engineers still experience a disconnect between how the overall Information Security Program affects their daily activity on the front lines of defense. This is the same problem that software engineers who formulated the Agile Credo came together in the early 2000's to solve.

How can we write better code? How can we produce more meaningful features? How can we deploy faster and with less bugs? How can we make our applications more resilient and reliable? It was not about the languages, or the frameworks. It was about the "how".

Thus, we can draw many parallels to the cyber operations engineers and software engineers. Both groups operate at the lowest level of their respective functions. But are we asking the same questions in the cyber context? How can we program better defenses? How can we enhance our existing toolsets to best protect the organization? How can we make changes faster and more continuously without breaking existing systems? How can we provide more fluid resiliency and respond to change quicker? There is no framework for this in the security operations space.

## The "Middle" Age of Security Operations: The Agile Blue Team?

The blue Team is the lowest common factor of a cyber defense program. Blue teams are responsible for the tactical implementation of the cyber program at the lowest level. In military terms, the blue teamers would be the front-line troops on the cyber battlefield. Blue teamers are responsible for the tactical defense programs; the identification, detection, and protection against cyber threats. The team also plays a key role in responding and recovering from cyber incidents. The blue team typically sits below the overall cybersecurity program, which, in military terms, typically looks like the "central command" equivalent of the cyber battlefield.

While most Information Security Programs are particularly adept at governance, risk, and compliance related techniques, there is no singular guideline as to how to operate a blue team. What roles should be defined? How should the teams communicate? Whom should they be communicating with? How do they prioritize and operationalize the security work that needs to be done? What methodology should they deploy to ensure the most attention is focused on the highest areas of risk to the organization?

The Agile credo created a pathway for the structured development of Agile methodologies to take hold amongst development teams and also helped to proliferate the engineering of DevOps principles and the subsequent tooling like CI/CD platforms. By contrast, the cybersecurity field lacks such a high-level credo. What are the core principles by which blue teams should be operate? What are the high-level drivers which define the operational tempo of an organization's cybersecurity program?

In absence of an "Agile Blue Team Credo", there does exist a basic cybersecurity "ethos" which may serve to define the tactical defense strategies employed by most security operations groups. The guidelines below generally summarize these core principles.

- Extreme focus on identifying and eliminating "residual" risk. Residual risk is a measure of risk in a system or asset after the compensating and mitigation controls are considered. Security Operations efforts should have an extreme focus on protecting systems that have high residual risk. This risk should be driven top down from the Information Security

program and the front-line engineers should clearly understand which systems the highest residual risk in order have to better understand where to focus their efforts.

- Continuous resiliency as a mantra. Attack the key areas of residual risk continuously. Prioritize and defend. Then do it again.

- The home field advantage**.** Cyber defenders should have an expert understanding of the environments in which they operate in. This is a key element to orchestrating defensive cyber tactics and techniques (honey-potting, logging, alerting etc.)

- Small unit tactics over bureaucracy and frameworks**.** Small, self-organizing, collaborative teams are better at identifying the keys areas of risk to an organization than higher-level frameworks and point-in-time risk assessments.

- Good security relies on sound architecture**.** Address underlying architectural changes as a primary consideration for mitigating risk. This requires Security Operations to "see the forest through the trees" so to speak. Blue teams need to understand how to architect security-by-design strategies.

- Structured process for continuous improvement, communications and collaborations. Build structured communications and collaborations frameworks for addressing cybersecurity risks in a systematic and structured fashion. Visualize your work and make it accountable. Limit work in progress and reduce constraints.

## People, Process, and Technology Agility: The "How" of Security Operations

Identifying and adopting these high-level principles allows organizations to shift mindsets in terms of how they think about blue team operations. Organizations wishing to increase their security posture must utilize small unit tactics as part of a larger governing framework. Ultimately the goal here is similar to the software credo: break down communications barriers and attack the areas of risk that are most critical to the business, separately prioritize cyber issues, and build your teams in such a way that enables the continuous resiliency concepts.

### People

As noted previously, the key roles in Agile development teams are very clearly defined: End Users → Product Owners → Scrum Masters → Developers → QA Engineers → Back to End Users. But what does an "Agile Blue Team" look like? As the cyber industry continues to progress, the roles that constitute blue teams are beginning to emerge. Perhaps then, a better question is what should blue teams look like?

If the prioritization of security operations is directly dependent on risk identification, then the first key role in the cyber organization should be the business systems, IT asset, and application owners. These key roles are akin to the end users in the software development world. They are the most qualified to help identify risks to critical applications and ultimately are

responsible for the daily security posture of the organizations key applications and systems. This group will have the tall task of conducting risk assessments on the applications, assets, and systems to which they are assigned. These subject matter experts should have a clear understanding of the threat landscape with regards to their own functional areas and be able to articulate the likelihood and the impact of an exploit on a given threat. Therefore, this group is the primary resource to an organization when deciding where to deploy their often-limited Security Operations resources.

The next key functional group is the Cyber Analysts. This is the group ultimately responsible for the prioritization and the planning of all the compensating and mitigating controls based on the threat assessments of the first line of defense groups. This group is responsible for overlaying the risk profile of the organization on top of the scanning and vulnerability management programs in order to better prioritize which security items should be prioritized and addressed first. In addition to vulnerability assessments, this group should also be the primary interface with external pen testers. This pairing of the analysts with your tactical red-teamers gives the red team high-value insight into which areas of the testing are the most impactful to the business. Conversely, the red team is able to provide real-time feedback on areas of the security posture that can be actively exploited. In mature organizations, the Cyber Analysts would also be responsible for gathering the threat intelligence and overlaying the data on top of the risk assessment frameworks.

Once the organization has a good structured approach to understand risk, and the subsequent criticality of that risk, this information should be fed down the assembly line to your "scrum masters." These are the blue team "cyber scrum masters" who are ultimately responsible for the prioritization of the work that needs to be done. This group manages the tempo of the security operations and ensures that resources are expended in the areas that require the greatest risk mitigation.

Once the work is more clearly defined and prioritized, the cyber engineers can focus on the work itself. This includes patch management, vulnerability management, threat hunting, and incident response. This group is akin to the actual developers in the software lifecycle and are responsible for actually deploying the compensating and mitigating controls across the applications and assets in the organization. This group is also tasked with instituting security by design approaches across the organization's architecture.


**Process**

Unlike higher level frameworks which operate at the organizational level, the agile principles of cyber operations define how organizations should operate at the lowest level. Take, for example, vulnerability management. While NIST helps to define how an organization should define its vulnerability management program, it does little to provide guidance the blue team engineer actually remediating the vulnerabilities on a daily basis. Most engineers know that vulnerability management is a Sisyphean effort. Typical vulnerability scanners such as Nessus, Qualys, or Nexpose will identify far more vulnerabilities in a network than can reasonably be remediated. These vulnerabilities are often described in the following terms (depending on severity):

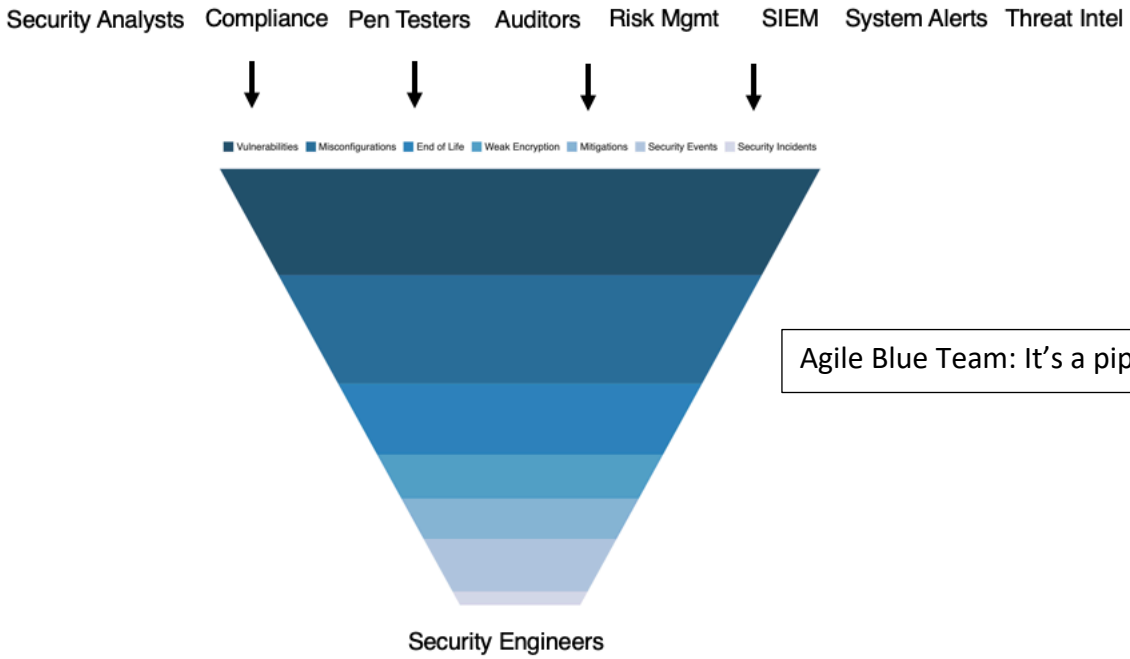
- Critical
- High
- Medium
- Low

Most engineers also know that all vulnerabilities are not created equal. A simple example would be a critical vulnerability on an externally facing asset should logically be treated with greater urgency than if the vulnerability was identified on an internal asset. In addition, a critical vulnerability in an asset that processes critical customer data should be treated with greater urgency than an asset that serves a less critical purpose. The Agile Blue Team, therefore, should have a standard methodology in which to treat these vulnerabilities. In other words, the organization's risk profile and risk methodologies should be overlay on top of the vulnerability set so that the engineers can separately prioritize and tackle the most impactful changes. This creates agility.

There are many other areas in Security Operations that would benefit from Agile-like methodologies. One emerging area where this is already occurring is the concept of "Purple Teaming" or "Continuous Pen Testing." In typical Agile fashion, blue team engineers work closely with their red team counterparts in order to shorten the cycle of improving cyber posture.

As red teamers identify areas of exploit, they partner with their blue team counterparts in real time in order to find ways to eliminate those exploits. This is the functional equivalent of how a software team defines the end-user feedback loop. The red teamer provides vulnerabilities and exploits in real time and the blue teamer remediates in lockstep.

Security Operations groups might also benefit from Agile principles by learning to better catalog and prioritize inbound security events. Blue teams are typically appendages to an already existing network operations group. Network operations groups already follow structured frameworks for cataloging and prioritizing inbound incidents or requests utilizing ITIL frameworks[4]. These frameworks, however, do not include categorizations for security related events or incidents. Tracking security events, and security incidents should be a natural extension of the ITIL framework. This would create a prioritized pipeline for security operations to function within and provide measurable, and repeatable processes.



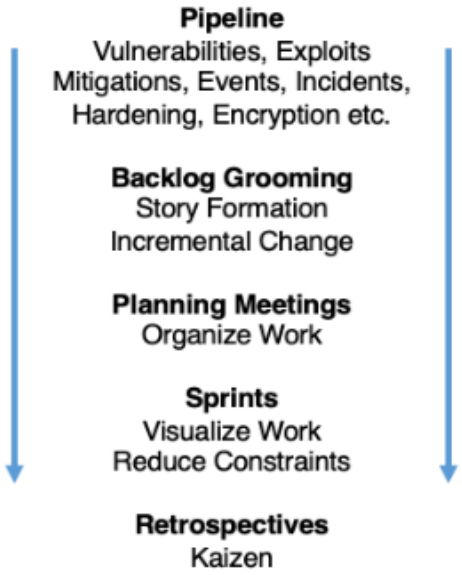

Agile Blue Team: It's a pipeline problem

When analyzed, the processes underpinning how blue team operations should function look very similar to how software teams view their pipelines. The only difference is that the inputs coming into the process are not features, bugs, or new requests from end users. Instead, they are the daily security related issues facing the security operations group. The underlying processes all look the same. It stands to reason that blue teams might benefit from the same methodologies by which software teams operate. This is where security meets agility.

---

[4] "ITIL." In *Wikipedia*, March 7, 2020. https://en.wikipedia.org/w/index.php?title=ITIL&oldid=944336671.

Agile Blue team: Process




**Technology:**

No organization's cyber posture would be considered agile without the use of technology and tooling to automate the program. Just as DevOps was the tactical embodiment of the Agile credo, we see similar trends underpinning the security operations programs.

One of the greatest challenges facing security operations groups is the difficulty in deploying secure configurations across all the applications, assets, and systems. Tools like the Center for Information Security (CIS) help to define the most secure configurations a range of assets such as desktops, servers, databases and applications. However, deploying these changes across the fleet of assets (in particular in non-homogenized IT environments) can be particularly challenging. Modern tooling such as Chef, Ansible, and Puppet seek to solve this problem in an automated way. For security operations groups, these tools are capable of managing and orchestrating these security configurations in a fully automated fashion. This "configuration-as-code" is a modern approach to security automation and is a direct parallel to what DevOps brought to the Agile technology stack.

In addition to continuous configuration management, there is also tooling available for verifying continuous compliance with these standards. These tools sit on top of your continuous configuration systems to verify that your fleet of assets conform to the configuration standards of your organization. These modern approaches to security mirror the approaches adopted by the development teams and the basic principles of DevOps such as Continuous Integration and Continuous Deployment.

Perhaps there is no greater area of parity to the Agile methodologies than in the area of DevSecOps. Security groups are now being embedded directly into the software development lifecycle in an effort to move security practices as far "left" on the SDLC as possible. This partnership between development and security teams embeds secure technology automation into the development lifecycle – thereby benefitting both functional groups.

DevSecOps is a natural extension of the DevOps which focuses on the automation of security principles as part of the software design, build and deployment processes. This is sometimes also referred to as "Security-by-Design" and basically adds security focused automation tooling and processes around the software development lifecycle. Prior to code being deployed into production it can be statically analyzed for potential vulnerabilities. In addition, upon deployment, code can be dynamically tested for additional exploits prior to making its way into a production environment. This pushes security further left on the stack to ensure the applications remain even more secure.

Some examples of DevSecOps tooling are things like Veracode of Checkmarx which analyze code repositories for common vulnerabilities before they are even deployed to testing environments. Further along the development lifecycle, security teams can partner with developers to deploy dynamic security testing which identifies "vulnerabilities-in-motion" by dynamically testing code for security vulnerabilities as part of the developers automated testing practices. If code fails on security testing (e.g. if a SQL injection or cross site scripting vulnerability is found) then the build and/or deploy will also fail. This prevents bad code from moving to production environments.

Security automation tooling is still a growing discipline. We now see automated endpoint detection and response in systems such as Crowdstrike e.g.. We see a plethora of AI based tooling both at the network and endpoint level. These tools aim to "make sense of the noise" and find signals in the data to eliminate the human layer. This will no doubt be an area of strong growth over the next few years.

## The Future of Continuous Resiliency

As the information security discipline continues to mature, we can draw on some of the lessons learned by the software development community. The dawn of the dot-com era saw explosive growth in the internet, and this required a new way of thinking. It required a systematized approach on the "how" of writing software. In the 20 years since then, there have been vast improvements in the software development people, process, and technology stack. However, the dawn of the cyber-age is just beginning. Forward leaning cybersecurity programs will need to adapt to a modern approach to protecting the organizations assets from cyber-attacks. Information security programs will need to reach further down the stack and include a tactical and technical approach to security operations that aligns with the overall strategy of the Information Security program. This involves more than just defining a strategy of continuous resilience. Instead, it requires that the Information Security program embed agile practices across the people, process, and technology stack.