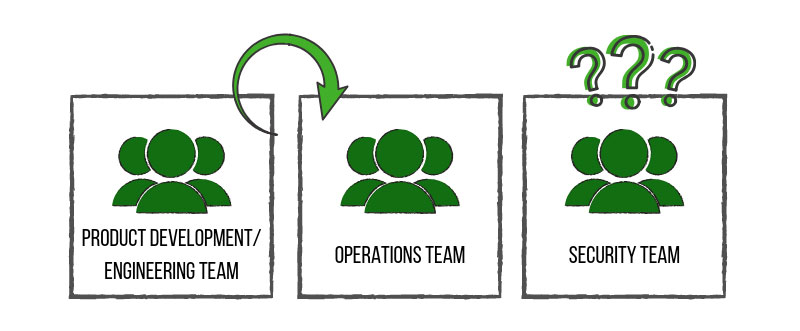


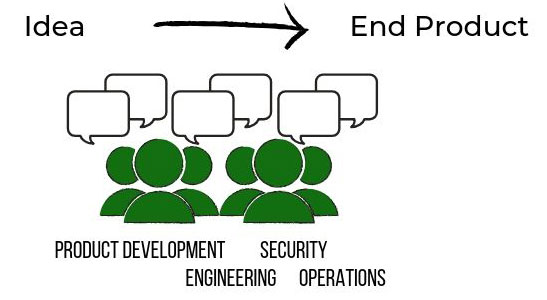
What is DevSecOps?

DevSecOps is the evolution of the DevOps philosophy. It is a concept that injects security into the software development lifecycle. If DevOps is about increasing the level of communication between development and operations, then DevSecOps is about inviting security into the conversation.

Before there was DevOps, organizations would divide work and communicate internally, but hardly ever between teams. When one team completed their tasks, they would pass it to the next team, throwing it over the proverbial “wall”, assuming that their job was done and they had nothing else to do with the project. The security team was often clued in at the end of the project, much like an afterthought. This lack of communication caused confusion and conflict between teams, slowed down production time and introduced more vulnerable products to the consumer. This, of course, consequently affected the path of value for companies.



When the DevOps methodology came about, it revolutionized the way teams worked together, with the goal of getting product and development to talk to operations. The advent of DevSecOps placed security into the big picture and allowed all involved teams to work together from development to end product – a stronger, more efficient and more *resilient* end product.



Read more about DevSecOps

• [**Devops and Separation of Duties**](https://newcontext.com/devops-and-separation-of-duties/)

• [**4 Tips for Building Secure Software**](https://newcontext.com/4-tips-secure-software/)

• [**Read all DevSecOps related blog posts**](https://newcontext.com/category/devsecops/)

Case Study

[Reducing Risk and Cost](https://1o94re31z2r53ujw5331m3or-wpengine.netdna-ssl.com/wp-content/uploads/2018/03/NC-Case-Study-Reducing-Cost-Risk.pdf)

DevSecOps Services

Get fast, secure and optimized delivery of highly scalable applications.

[**Learn More**](https://newcontext.com/devsecops-infrastructure-automation-orchestration/)

How do you implement DevSecOps?

If you are pushing for digital transformation and looking to mature your engineering practices, implementing DevSecOps from no foundation could be daunting. New Context has helped many companies take those giant steps. Here are some of the things that we have learned:

Don’t Try This Alone

If you have little to no experience or haven’t been part of another DevOps team (or its equivalent), get to know people who have. The DevSecOps community is vast, supportive and constantly growing. You can begin with online resources ([we recommend DevOps.com](https://devops.com/category/blogs/devsecops/)) to better understand the concepts. Whether you come to us or reach out to another organization make sure that you take steps to [get more acquainted with the broader DevSecOps community](https://www.meetup.com/topics/devsecops/).

Iterate and Test and Test and Test…

As you build out your new software pipeline, testing becomes the core piece that protects your infrastructure. Learn to see build failure as a good thing, rather than a setback. Each failure reveals an opportunity to learn, which allows you to build a stronger product. The stronger your tests the stronger your software, and the more secure it can potentially be.

The Automation of Trust

The implementation of security and compliance automation reduces overhead involved in the management of your software and infrastructure. Look to see how you can build a process to implement your security policy as code. Take your compliance controls and build them into your release pipeline. This increases both efficiency and consistency and will reduce the risk of introducing potential security flaws, making it a more trustworthy product.

Communicate, Communicate – And When You’re Done Communicating, Communicate Some More.

Making an organizational shift is not easy. There are a lot of complexities for running and putting together a new way of doing business. Each of your team members will also mature and approach the transformation in different ways. The key is to make sure everyone walks the path together. A regular cadence of Standups and Retrospectives can go a long way to helping your organization stay connected. But be mindful when using digital tools, as they may hinder the delivery of voice inflection, eye contact and mood, all of which are vital during critical discussions.

Have a Principled Approach

Have a principled framework that works for your organization. This will allow your people a constant reference for the work they do. [Our Lean Security Manifesto](https://newcontext.com/leansecurity/) outlines the 4 principles that we adhere to for keeping everyone focused: Awareness, Simplification, Automation and Measurement. Having these 4 core values top of mind has enabled us to become consistent in the solutions we provide for our customers no matter who in our company is engaging with them.

Feedback Loops

Build constant feedback loops that give you viability in the process. Make sure that you track and analyze the key performance indicators (KPIs) that determine your success. Once you’ve finished analyzing, make the necessary adjustments to improve your product or project. Rinse and repeat.

Why DevSecOps?

Increased Velocity

With proper DevSecOps implementation, the automation of processes will not only allow you to develop more efficiently, but it will also strengthen your software, which means a better product reaching the market faster. A better product equates to happy customers, ensuring your ability to compete in the market.

Reduced Risk

Early implementation of compliance and security ensures a better code base and stronger security posture. Taking a more proactive approach to catching bugs and defects decreases vulnerability. You will also be able to respond to incidents significantly faster if you begin developing your product with the risks in mind.

Simply put, the early integration of security tooling into the software development process ensures a better product. When running automated tests, security tests are also run. When new features are in the design phase, security questions are asked, such as:

* Is this feature going to attract bad actors?
* Are there people that will want to do bad things with our software if we allow this good thing to happen?
* If so, how can we prevent that?

Security needs to be closely involved from the “aha” to the “cha-ching”.

In the end, incorporating security early in the process can reduce headaches and budget overrun.

Key Performance Indicators (KPIs) of DevSecOps

Release Cycle Time

Code deployments should happen more often, and complete in less time. Shorter cycle times are a sign of optimized processes, while longer times could be a sign that you need to reexamine your coding best practices and/or tools. Start tracking your cycle times to get a baseline in place, and then begin measuring variations in that baseline to see how things change.

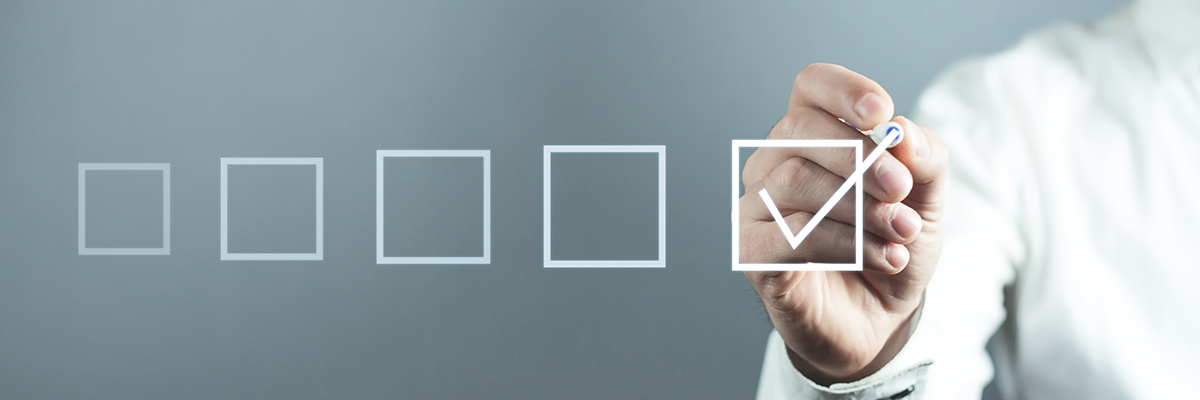
Code Quality and Risk

Keep a close eye on customer-initiated contacts, bug reports, and code coverage percentage. The risks of low-quality code include performance and security risks, which can be costly to both the company and the consumer. Good quality code is consistent, well documented, and able to be tested.

Development Time-to-Market (TTM)

Code should make its way through development faster, from ideation to deployment push. Review your process and see if there are any unnecessary meetings or duplication of efforts that could free up more time for innovation and testing. Ensure communication between teams is streamlined, and remember to automate whenever possible. A faster time-to-market means greater competitive advantage for your company.

4 Tips for Building Secure Software



Software doesn’t write security in by itself; it takes best practices and intent to write good secure software. The industry is starting to agree on this, and the talk by [Kelly Shortridge and Dr. Nicole Forsgren at Blackhat 2019](https://www.veracode.com/blog/security-news/live-black-hat-usa-inevitable-marriage-devops-security) about bringing InfoSec and DevOps together is a great addition to the discussion. In the spirit of sharing more knowledge, here are some tips to help you and your fellow software developers build with more security in mind.

**1. Understand Your User**

Before writing a single line of code, you must know the “Why” behind your software. What are your user’s needs? How will they use the end-product? Without a clear vision of your ideal user, it’s hard to create a solid framework. Make sure that you have done proper research to answer these questions confidently.

**2. Know Your Risks**  
***Security***: Start thinking about security before software development has even begun. Continue thinking about security requirements throughout the development and deployment process. Stories, improvements, new features should all include a definition of “Done” that is inclusive of security requirements.

* ***Quality***: Measure the code quality over time. Benchmark your metrics and provide updates, allowing everyone to see how things are progressing. If you need more information on choosing your metrics, check out [O’Reilly’s article on code quality.](https://www.oreilly.com/ideas/code-quality-and-measuring-what-matters)
* ***Prudency***: If a system is experiencing issues, how will it affect the user **and** the bottom line? It is rumored that if a customer signs up for Netflix and their credit card payment system is down, Netflix just gives them “*X*” number of free days of their service. It’s better to just let the customer stream now and figure out payment later. Netflix realizes that they aren’t losing much money to a few users streaming for free for a few hours.

**3. Always Use Coding Best Practices**  
These practices include always checking your inputs, having a process in place to prune your code on a regular basis and using TDD and static analysis processes. You should also set up your new features behind feature flags for more control.

Consider using an automated pipeline that does security checking for you, and set it up to run checks *all the time*(not just specific points in time). Refer to the GSA [Information Technology Policy](https://www.gsa.gov/policy-regulations/policy/information-technology-policy) or NIST [Cybersecurity standards](https://www.nist.gov/topics/cybersecurity) for more information on methodologies to protect your users.  
And of course, make sure engineers get immediate feedback on their code. Be able to answer the following questions:

* Does it have bugs?
* Is it written poorly?
* Is it written with security problems?
* Does it follow policy?
* Has it been peer-reviewed?

**4. Set Your Engineer Partners up for Success**  
***Train***: Whether you are a manager or an engineer, make sure your fellow engineers have an adequate amount of time and resources to learn the skills and tools needed to build the software. This is a marathon, not a sprint, and good practices take consistent work — consider the athletes who are always training. All professions need consistent work to be better.

***Strong Communication Tools and Practices:*** Every engineer should understand the company’s mission and vision. This ensures that they take these values and apply them to their work. When a team knows a company’s purpose, its stakeholders, and how it intends to accomplish its goals, staying on track is that much easier.

***Prepare for The Future***: Make sure your team knows what may be audited or part of an upcoming compliance check. Knowing compliance requirements, what to expect during an audit and having solid software asset management (SAM) procedures in place can prevent any unsavory surprises.

The key to building more secure software is to consider every angle at the start of your project–including (especially!) security. Ask the right questions, know your team’s capabilities and your user’s needs. The InfoSec landscape is constantly evolving; keeping an eye on its growth, and developing a strong interest for building security into your software is critical to your success.

To get an idea of how your organization is doing, try our [LS/IQ DevSecOps score card](https://newcontext.com/lsiq-lite/); it may give you some insight to how mature you are in this process today.

https://newcontext.com/what-is-devsecops/