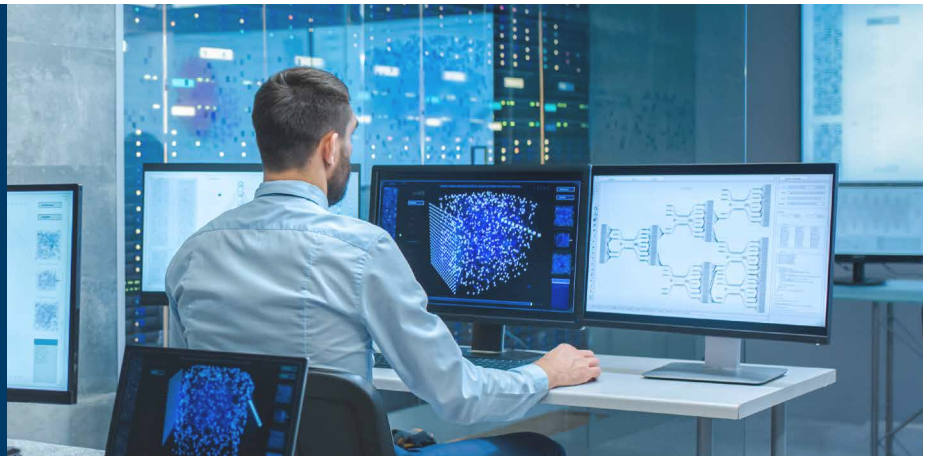
The Institution of Engineering and Technology

# DevSecOps in a safety-related environment

This flyer on DevSecOps is one of two high-level IET guidance documents on the benefits and pitfalls/challenges of Agile development and DevSecOps practices for organisations undertaking safety-related projects, products, or services. Incorporating security into software development is essential to counter increased cyber threats.



DevOps (a portmanteau of development and operations) and its extension, DevSecOps (a portmanteau of development, security and operations), aim to break down the traditional organisational silos between teams responsible for each of these areas and produce software in a holistic and highly-integrated way. In doing so, it reduces costs, increases delivery rates and enhances a culture of transparency and assurance. The flyer's aim is to spread awareness and to increase understanding and good practice amongst engineering managers in this area.

DevOps sets out to achieve its promise of delivering higher-quality software and shorter development cycles through a combination of pre-existing concepts for managing, delivering, and monitoring software, such as:

### Extensive use of version control systems

Version control systems serve as a configuration-controlled repository for code and have formed a part of best practice in software development for many years. DevOps emphasises the use of their collaboration features and other functionality to accelerate development.

### Continuous integration (CI)

The process by which software builds are produced and tested as the software is updated. Automatic testing provides fast feedback and metrics to developers and project managers respectively and allows test effort to be focused on those aspects which cannot be automatically tested.

### Continuous delivery / deployment (CD)

The process by which software can rapidly and automatically be configured and deployed, to stakeholders and users, with minimal interaction.

### Infrastructure as code

This concept ensures that the infrastructure supporting the development, testing and delivery of software is managed as though it is software, benefiting from the other concepts.

### Monitoring and logging

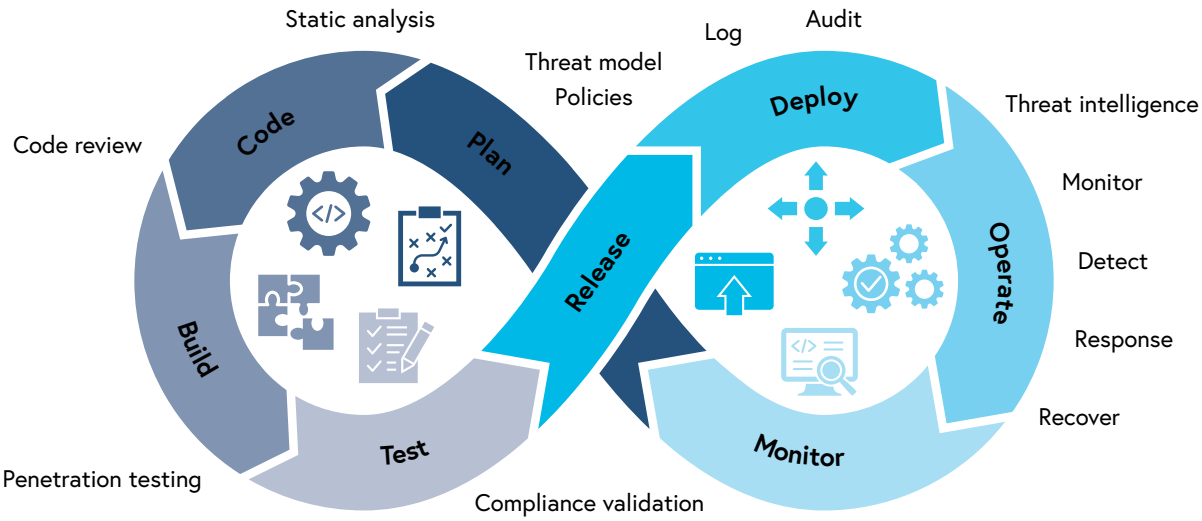
Once software is deployed, logging and monitoring can be used to capture metrics on performance, as well as identify any potential issues before users report them.

**DevSecOps** is a further development of DevOps which seeks to integrate security activities, such as:

1. Undertaking **vulnerability scanning** prior to major software releases to identify potential security vulnerabilities long before the software reaches users.
2. Carrying out **security testing** before software is released, using a database of reported exploits and security flaws, ensuring that security flaws are not reintroduced.
3. Software that can be **architected for security** from the outset, ensuring that concepts such as encryption and zero-trust can be fundamental parts of the software architecture, design, and implementation.

The Institution of Engineering and Technology

**Figure 1: DevSecOps lifecycle.**




Safety standards rely heavily on the need for software to be traceable and auditable to ensure that the software meets its original requirements and has not undergone uncontrolled modifications or changes. DevSecOps represents a paradigm shift for organisations working in this environment as the process of developing and assuring software can be done incrementally while enabling all changes to be traceable. Equally, DevSecOps requires that organisations identify all dependencies (such as external libraries) needed to successfully build software, enabling these to be assured from both a security and safety standpoint. Safety standards also increasingly stipulate that cyber security assessment and activities are carried out; DevSecOps can support this objective.

The adoption of DevSecOps approaches should therefore be considered by organisations working within the safety-related environment, assuming that the following questions can be answered:

**Does the organisation find itself often taking significant periods of time to produce software builds?**

**If yes**, the automation provided through the application of DevSecOps principles and concepts can represent a significant time saving and accelerate the feedback cycle.
**If no**, DevSecOps can still provide an automated method for the production of builds, resulting in overall higher build quality and frequency.

**Does the organisation have a significant body of tests which must be executed whenever the software is changed?**

**If yes**, DevSecOps pipelines can be configured to apply a build verification test to any modifications to the code while full regression can be deferred to convenient times.
**If no**, DevSecOps can still provide many tools to support a higher level of test automation and orchestration, which enables manual verification and validation activities to focus on complex tests.

**Does the organisation have formally defined security obligations, e.g. as part of a regulatory regime?**

**If yes**, tests to verify these formally defined security requirements can be created and executed as part of the DevSecOps pipeline processes.
**If no**, generic threat modelling techniques can still be used, potentially supplemented with automation, to identify security related non-functional requirements and potential test cases for consideration.

DevSecOps enables organisations to combine much of their software build, deployment, testing and monitoring capabilities while additionally integrating cyber security principles. This represents an opportunity in a safety-related environment, as rich provenance and traceability will be established for all components involved in producing the software build, supporting the assurance case. High degrees of effective automation enable teams to focus on other key activities, such as the reduction of technical debt and any residual manual testing. DevSecOps therefore represents a method for accelerating software development while also providing benefits to the assurance case.

This is the first in a series of IET outputs on this topic. A more detailed document is currently being developed and will be published shortly.

**All feedback on this paper is welcome. Please contact sep@theiet.org. This paper has been produced by the Engineering Safety Policy Panel. For more details on the Panel's work, visit theiet.org/engineering-safety.**



The Institution of Engineering and Technology (IET) is working to engineer a better world. We inspire, inform and influence the global engineering community, supporting technology innovation to meet the needs of society. The Institution of Engineering and Technology is registered as a Charity in England and Wales (No. 211014) and Scotland (No. SC038698). Futures Place, Kings Way, Stevenage, Hertfordshire, SG1 2UA, United Kingdom.

E6D22001/Engineering Safety/0722