automate | monitor | optimise

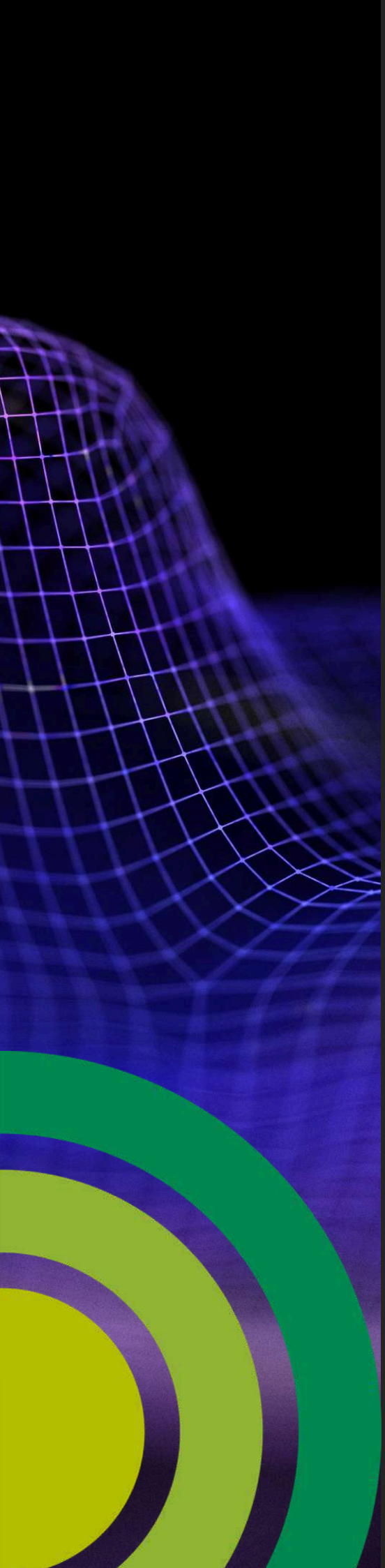
avocado

Avocado Consulting

DevSecOps Maturity Model

A guide for evaluating and progressing your organisation's DevSecOps initiatives

deliver with certainty



avocado

Contents

PAGE 3. INTRODUCTION

PAGE 4. SECTION ONE: ASSESS

ASSESSING THE CURRENT STATE OF DEVSECOPS MATURITY
Four Pillars of Maturity Model

PAGE 8. SECTION TWO: DEFINE

DEFINE THE DESIRED MATURITY LEVEL
A look at the Maturity Levels

PAGE 10. SECTION THREE: STRATEGISE

DEEP DIVE INTO DEVSECOPS USE CASES

EATING THE ELEPHANT
Avocado's approach and methodology

PAGE 16. CONCLUSION

deliver with certainty



"The purpose and intent of DevSecOps is to build on the mindset that everyone is responsible for security with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required".

- Shannon Lietz, co-author of the "DevSecOps Manifesto"

avocado

Secure software supply chains

Delivering fast, safe, and secure digital services is crucial for successful transformation, yet for many, current organisational practices lag behind.

Modern software development revolves not only on Agile, but secure software supply chains.

Prioritising security earlier in development is key. To efficiently deliver secure services, organisations can improve their maturity by adopting DevSecOps methodologies. This holistic approach speeds up IT delivery while maintaining controls. To do so, leaders must:

1. **Assess** the current state of DevSecOps maturity.
2. **Define** the desired maturity level.
3. **Strategise** the transition path to future state.

This whitepaper introduces DevSecOps maturity concepts and builds understanding of expected competencies, guiding self-assessment and transition. It aims to advance from 'adhoc' to 'elite' practices, to showcase DevSecOps' business value.

As experts in empowering clients to develop internal DevSecOps capabilities, we delve into Avocado's proven methodology, approach, and framework, enabling organisations to uplift DevSecOps from zero to hero.

Section one



Self-assess your maturity

Assessing the current state of DevSecOps Maturity.

Understanding your DevSecOps maturity is crucial for identifying strengths and weaknesses, managing risks, allocating resources effectively, ensuring compliance, and maintaining a competitive edge. It provides a baseline for improvement and helps in setting realistic goals for enhancing security practices throughout the software development lifecycle.

How mature is your DevSecOps?

Avocado have designed a DevSecOps Maturity Self-Assessment tool to quickly gauge your maturity level, from Adhoc to Elite. Aligned to industry best-practice, it sets you on a path to improve your DevSecOps maturity.

Our self-assessment tool is specifically designed to get an understanding of your current environment. It asks **20 multiple-choice questions** gauging expected maturity level competencies across foundational pillars allowing you to quickly check your current practices, identify areas for improvement, set realistic goals for advancement and then track progress overtime.

Take the assessment now

avocado
CELEBRATING 20 YEARS

FOUR PILLARS OF MATURITY MODEL

The four pillars of DevSecOps provide a foundation to develop security-first software within industry standard practices:



Avocado's DecSecOps Maturity Self-Assessment is based on four foundational pillars of maturity model. Each pillar is aligned to the Secure Software Development Framework (SSDF), a NIST industry standard framework that helps organisation to align and prioritise its secure software development activities with its business/mission requirements, risk tolerances, and resources.

DevSecOps excellence requires early involvement of security teams alongside development and operations to seamlessly integrate security into the development pipeline. This necessitates a shift towards a shared goal of delivering fast and safe software - prioritising collaboration, automation, and continuous improvement across the organisation. These can be broken down into four pillars, defined on the right.

Maturity levels depend on core competencies that are expected against each of these pillars such as: policies and procedures, roles and responsibilities, training, security tools, identity protection, secure by design principles, audit and assessments, and vulnerability management. We explore these pillars further.

1. Protect your organisation – Ensuring organisational readiness of people, tools and process.
2. Protect your software – Incorporating robust security measures at every stage of the software lifecycle.
3. Produce well-secured software – Security "Shifting Left" so it's integrated into the requirements and design of the software.
4. Proactive response to vulnerabilities – Implementing Observability tools and strategies that allow for an appropriate response to vulnerabilities.

Exploring the four pillars further

Pillar 1: Prepare your organisation

Preparing the organisation emphasises the importance of readiness at all levels - organisation-wide and within individual teams - to carry out secure software development, across people, processes and products.

People: Ensure that all team members - from developers to project managers - are trained in secure coding practices and understand the importance of security in software development. This includes awareness of common security threats, understanding of secure design principles, and familiarity with security tools and technologies.

Processes: Secure software development processes should be established and followed consistently. This includes processes for threat modelling, secure code review, security testing, and incident response. These processes should be integrated into the overall software development lifecycle.

Products: The right tools and technologies should be in place to support secure software development. This includes platforms for security code analysis, vulnerability management, and automated testing. The product's technology stack should be kept up-to-date, security requirements tracked, policies in place with access controls.

At the organisation level, there should be a ‘security-first’ culture, and secure software development practices should be a standard part of the software development lifecycle. In some cases, individual development groups or projects may have specific security requirements that need to be addressed. Therefore, it's important to assess and prepare for secure software development at both the organisation level and the level of individual development teams. This comprehensive approach helps to ensure that your organisation is prepared to secure every aspect of software development.

Pillar 2: Protect your software

Protecting all components of software from tampering and unauthorised access involves implementing robust security measures at every stage of the software lifecycle, from design and development to deployment and management. It includes practices such as credential protection, secrets management, secure coding, regular security updates, secured code repositories, and the use of encryption to protect data. Additionally, access controls and authentication mechanisms are put in place to ensure that only authorised individuals can access and modify the software components.





Exploring the four pillars further cont...

Regular security audits and penetration testing are also conducted to assess the effectiveness of the security measures in place and identify areas for improvement. Ultimately, the goal is to create a secure software environment that is resilient to threats and can safeguard the integrity, confidentiality, and availability of the software and its data.

Pillar 3: Produce well-secured software

Producing well-secured software involves a comprehensive approach that starts from the design phase, where security is integrated into the requirements and design of the software. Secure coding practices are followed during the development phase to prevent common vulnerabilities such as buffer overflows, injection attacks, and cross-site scripting. Additionally, the use of up-to-date libraries and dependencies is crucial to avoid known vulnerabilities that could be exploited.

Rigorous testing is conducted before the software is released. The ‘shift-left’ approach ensures that security practices are integrated from the earliest stages of the development lifecycle. This includes secrets scanning, security testing including evaluating third-party libraries, and security audits to identify and fix potential security issues. After the release, continuous monitoring is essential to detect and respond to new vulnerabilities that may emerge. The aim is to discover vulnerabilities as early and fast as possible, but also to ensure a quick and effective response when vulnerabilities are discovered.

Pillar 4: Proactive response to vulnerabilities

There will be vulnerabilities that persist even after the software has been released, and they can pose significant risks if not addressed promptly. Appropriate responses to these vulnerabilities may include issuing updates to fix the vulnerabilities; notifying users of the vulnerabilities and providing them with guidance on mitigating risks; having robust incident management to plan and, responding proactively for risks and remediations.

Observability tools provide advanced AI capabilities such as risk prioritisation and intelligent automation with runtime vulnerability analytics and application protection. Preventing similar vulnerabilities from occurring in the future ensures learning from past vulnerabilities and improving the software development and testing processes. This could involve enhancing coding standards to prevent certain types of vulnerabilities, improving testing procedures to better detect potential vulnerabilities, or implementing better security controls in the software architecture.

Section two



A look at maturity levels

Define the desired maturity level.

The DevSecOps Maturity Model identifies four levels of maturity within the four pillars based on core competencies. In this section, we give an overview of each level and competency before presenting how to use the self-assessment results.

Reaching a certain DevSecOps Maturity Level is a journey that involves the continuous integration of security practices into the development process, requiring an organisational shift in culture, processes, and technology.

After understanding the current state of security, you must define the desired state or maturity. This then helps build an actionable roadmap that outlines the transition path from the current to the desired state of DevSecOps maturity.

Uplifting maturity involves transforming how an organisation approaches software development, emphasising security early on (shift-left). This enhances the organisation's ability to prevent, detect, and respond to security threats, advancing its DevSecOps Maturity Level.

When analysing assessment results and setting maturity goals, organisations should assess their current state and determine the necessary steps to reach their desired state. This involves identifying actions to bridge the gap and considering use cases that support integration, automation, and security culture in the software development lifecycle.

Next, we explore these maturity levels.

A LOOK AT MATURITY LEVELS

A prescriptive guide to measure the efficiency and efficacy on a specific domain:



Avocado's DevSecOps **maturity level chart** helps organisation's assess their current practices, identify areas of improvement, set realistic goals for advancement, and track their progress over time.

Adhoc (Level 1)

This is the foundation of DevSecOps. This level marks the beginning of the DevSecOps journey. Most important is a shift in culture and mindset that emphasise sharing and collaboration across technical disciplines, and a desire to improve performance as a cross-functional team.

Beginner (Level 2)

In this level, organisations start enforcing critical policies and security requirements and are shifting earlier in the development process, much of the security-related work is still done towards the end of the process. Security training is being implemented to equip your developers with a security-first mindset. At this level, your organisation starts leveraging security testing and scanning tools to detect exposed secrets and common vulnerabilities.

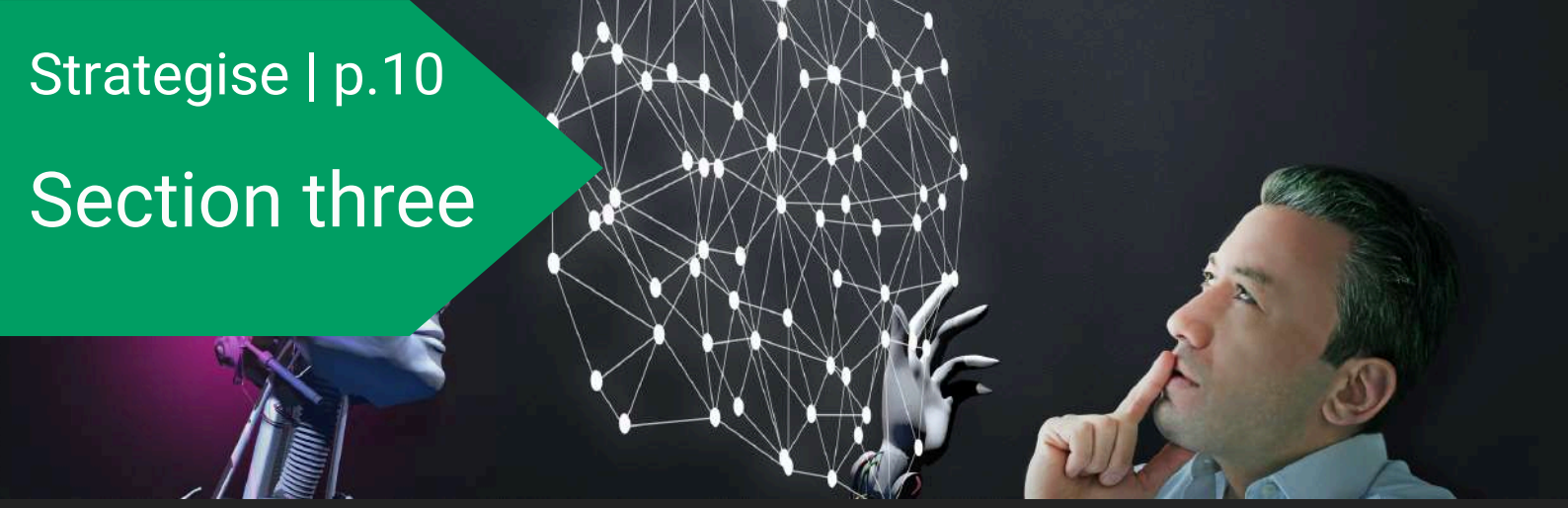
Advanced (Level 3)

In this level, organisations are highly efficient and productive, releasing high quality, secure software on a regular basis for a reliable product. Security checkpoints are embedded throughout the software development lifecycle, integrated in CI/CD, and vulnerabilities are captured as early as build phase. Application secrets are stored in centralised vaults and audited regularly. Third-party libraries are well managed, actively detected for security vulnerabilities. Incident processes are also well-established.

Elite (Level 4)

DevSecOps practices are well-established and mature at this level. Security controls are deeply embedded throughout the SDLC, and security has ceased to be a siloed domain. A key aspect of this stage of maturity, is a very high level of automation of processes across Development, Operations, and Security. Application runtime analysis and protection are detected real-time with the help of intelligent automation and Observability tools.

Section three



Strategise the transition path to future state.

Now you understand your current maturity what's needed on your path to DevSecOps maturity?

A deep dive into DevSecOps use cases reveals a wide range of solutions across different stages of the software development lifecycle to uplift your Maturity Level. Avocado have developed an iterative approach to developing use cases within your organisation through the following stages: Plan, Design and Build, Test and operate and, Monitor and respond.

In the Plan stage, Avocado Consulting offer solutions like Development Security strategy, Threat Modelling and Security Risk Requirements to identify and mitigate potential security risks.

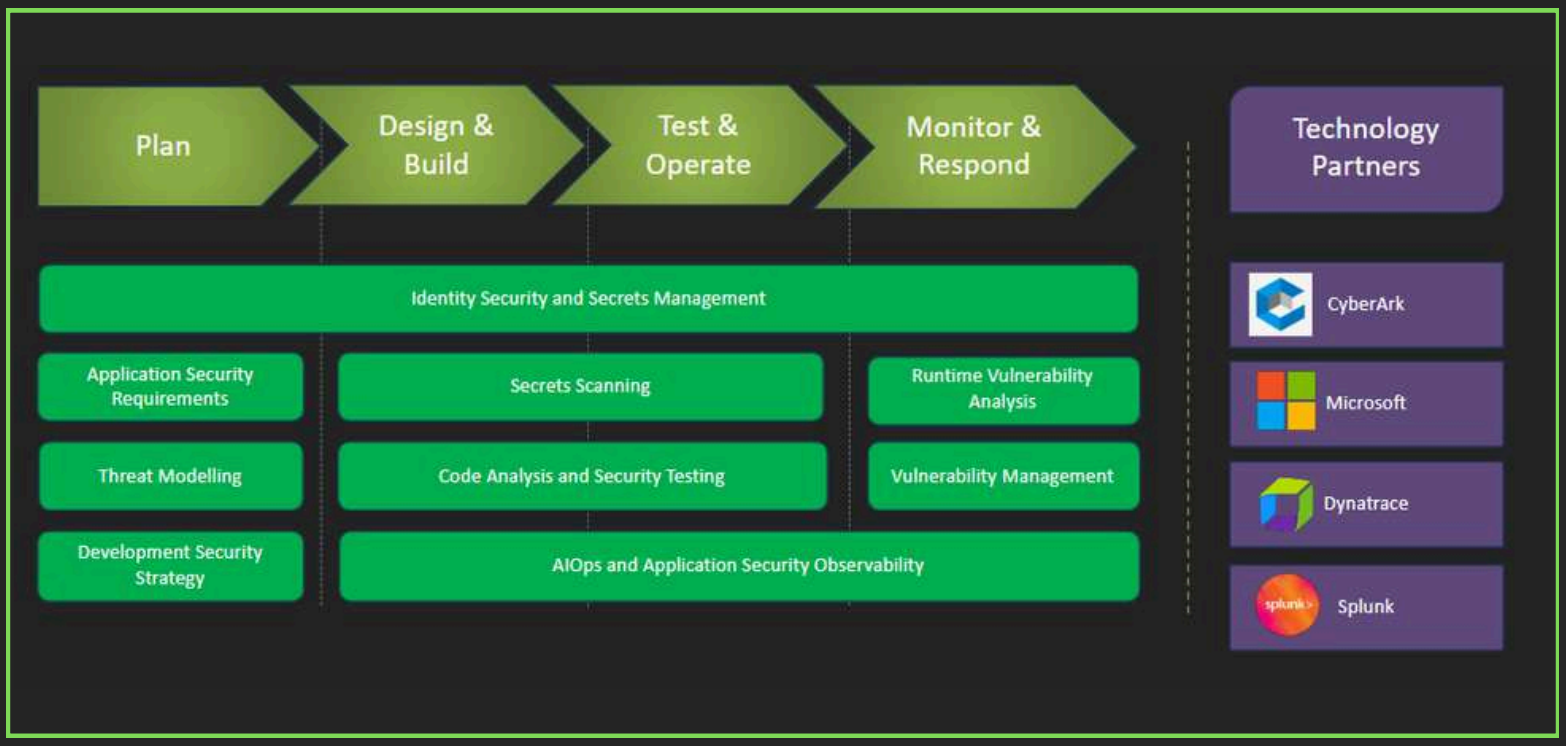
During the Design, Build, Test, and Operate stages, Avocado's solutions for Secrets scanning, Code analysis, Security testing and Penetration Testing, and Application Runtime Observability are introduced to automatically detect security vulnerabilities in the application code and infrastructure, whether in code repository or running in the environment.

In the Monitor & Respond stages, continuous monitoring and incident response mechanisms with security analytics and protection can quickly detect and respond to security threats.

Across all stages, Avocado's Identity Security solution enforces identity security controls from Privilege Access to Secrets Management to ensure your developer's identities and application identities are enforced with right privilege controls, centrally secured in a vault, and audit compliance in all your software development environments.

DEVSECOPS USE CASES

End-to-end Use Cases to Uplift your DevSecOps Practice



These above use cases illustrate how DevSecOps integrates security into every aspect of the software development process, enhancing the overall security posture of the organisation and delivering a secured customer experience. We explore each further below:

1. **Identity Security and Secrets Management:** Implementing robust identity management protocols and secure secrets storage to safeguard sensitive information access.

2. **Application Security Requirements:** Defining comprehensive security requirements at the outset of development to ensure secure coding practices and mitigate potential vulnerabilities.

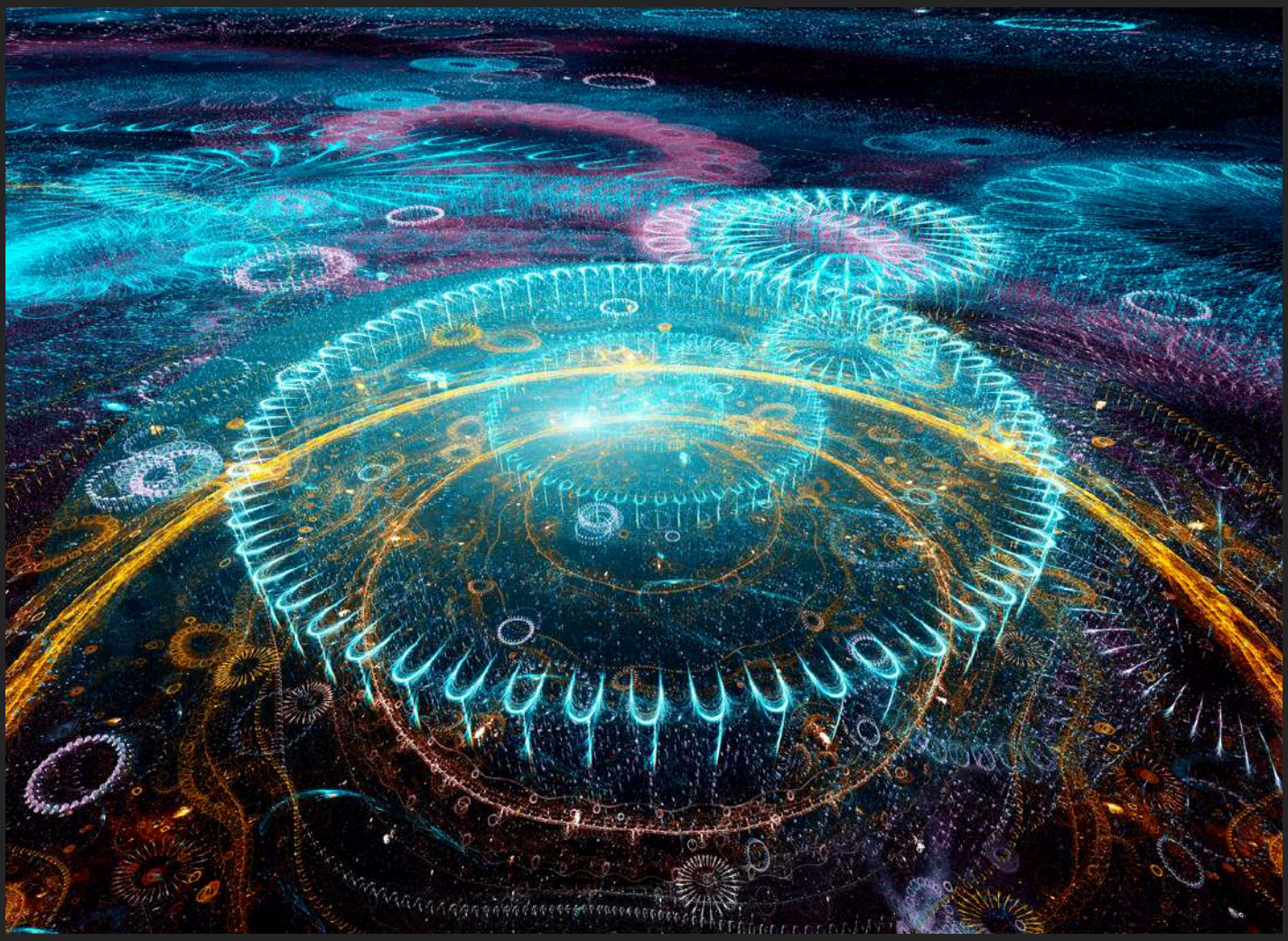
3. **Secrets Scanning:** Regularly scanning code repositories and environments to detect and remove hardcoded secrets and credentials, minimising the risk of unauthorised access.

4. **Runtime Vulnerability Analysis:** Continuously monitoring application runtime environments to identify and remediate potential vulnerabilities in real-time.

5. **Threat Modelling:** Conducting structured assessments to anticipate potential security threats and prioritise mitigation efforts throughout the development lifecycle.

6. **Code Analysis and Security Testing:** Employing automated tools and manual reviews to analyse code for security flaws and conducting thorough security testing to identify and address vulnerabilities.

DEVSECOPS USE CASES CONT...



7. **Vulnerability Management:** Establishing processes to track, prioritise, and remediate vulnerabilities promptly across applications and infrastructure components.

8. **Development Security Strategy:** Formulating and implementing a comprehensive security strategy that integrates security into every phase of the development process to proactively address security risks.

9. **Full-stack Context Monitoring (Observability):** Leveraging AI-driven operations and observability tools to gain insights into application behaviour and security posture, enabling proactive threat detection and response.

Technology Partners
Collaborating with technology partners like Splunk, Dynatrace, Microsoft, and CyberArk to integrate their specialised solutions into the DevSecOps pipeline, enhancing visibility, automation, and governance capabilities across the software development lifecycle.



Eating the elephant: Avocado's methodology and approach.

When navigating complex change, a structured framework keeps you accountable to your goals. Avocado offers established methodologies and frameworks for enhancing DevSecOps capabilities, focusing on roadmap creation and implementation planning.



APPROACH

Avocado's Assess, Plan, Do, Optimise (APDO) approach for the implementation of DevSecOps covering Avocado’s best practices in discovery and planning, implementation to optimisation, and expanding use cases.



01 - Assess
Pre-project to ascertain clearly business objective against the company's strategy and understanding current state to establish and baseline project scope.

02 - Plan
Work collaboratively with key stakeholders to develop a roadmap to determine prioritization of applications for iterative delivery, establish design patterns and assess organisational change impacts.

03 - Do
Develop and deploy iteratively making use of timeboxing to control the iterations.

04 - Optimise
Develop and deploy iteratively making use of timeboxing to control the iterations.

Build incrementally from firm foundations

Avocado’s Delivery Approach is a structured framework that requires minimum steps to design, approve and initiate the development of the solution. One of the development principles that we adopt at Avocado is to build incrementally from firm foundations.

To have a firm foundation, Avocado proposes to conduct discovery workshops where organisation requirements, use cases, objectives, success criteria are identified and reviewed. Then agreeing expectations on the technical solution, roadmap and timeline.

By applying our Software Development Framework to Avocado’s Assess-Plan-Do-Optimise (APDO) approach, we have been able to demonstrate value of solutions quickly therefore creating confidence and early adoption within our clients.

METHODOLOGY

Injecting security into the software development framework

ASSESS & PLAN

- DevSecOps Maturity Assessment
- Security Risk Assessment
- Discovery Workshops
- Roadmap and Benchmark Metrics
- DevSecOps Solution Blueprint
- Jump-Start Security Program
- Governance and Change Management



DO & OPTIMISE

- Security Implementation
- Security Standards and Best Practices
- Training and Documentation
- Security Automation and Workflows
- Operations Management
- Security and Development Team Alignment

Avocado's methodology provides set of success drivers to enable organisations to success. We guide organisations with state of the art Fast and Safe IT delivery supporting the core drivers of Practices, Collaboration, Culture & Process, Measurement and Technology. As highlighted in the graphic above.





Start your DevSecOps journey today

Conclusion: start your DevSecOps journey today.

With cyber risk increasing, being able to preempt potential breaches by enhancing both security levels and customer satisfaction adds value to an organisation.

By employing a DevSecOps framework, security seamlessly integrates into the development process, facilitating the incorporation of cost-effective security measures at the beginning of the software's cycle.

Maturing your organisation's DevSecOps requires a holistic approach that considers core competencies across the DevSecOps four pillars of maturity and core competencies within each of these pillars. Organisations must assess the current state of DevSecOps maturity, define the desired maturity level and strategise the transition path to future state, to ensure they can deliver fast and safe.

Don't know where to start?

Avocado offers a free consultation that sets you on a pathway to DevSecOps maturity, through business problem and use case identification.

Book a FREE Consultation



DevSecOps Services

Learn more about our DevSecOps Services aimed at enhancing your maturity

Automate, monitor and optimise your technology to deliver a superior customer experience.

Automate

Software Development | Software Testing | DevOps | DevSecOps

We use automation to lower costs, improve quality and speed up delivery. We reduce complexity and proactively resolve incidents.

Monitor

Cybersecurity | Observability | Application Monitoring | Managed Service

We monitor your IT environments to ensure performance, increase visibility and enhance your security posture which creates better customer experiences and governance.

Optimise

Product Delivery and Governance | Change Management | DevOps Delivery | ROI Improvements | DevSecOps Maturity Assessment

We optimise applications and infrastructure to lower operational costs. We streamline product delivery pathways to accelerate your speed-to-market.

avocado

For more information

contact us

ABN 14108342561

Sydney Office
02 8905 0198
Suite 10.05, 3 Spring St, Sydney NSW 2000

Melbourne Office
03 8640 9021
Level 5, Suite 5.05/454 Collins St, Melbourne

www.avocado.com.au