e2e assure

Application Security Maturity Assessment

Service Definition Document

Application Security Maturity Assessment

Service Definition Document – G-Cloud 15

CONTENTS

SERVICE INTRODUCTION

e2e-assure's Application Security Maturity Assessment (ASMA) provides organisations with a clear, evidence-based understanding of the effectiveness and maturity of their application security capabilities across the full software delivery lifecycle. The service is designed to support risk-informed investment decisions, regulatory readiness, and measurable security improvement.

SERVICE OVERVIEW

The ASMA engagement assesses governance, processes, tooling, and technical controls across applications, APIs, cloud platforms, and the software supply chain. Using a structured maturity model and consistent scoring approach, the assessment identifies systemic gaps, prioritises risks, and produces a practical roadmap and delivery backlog aligned to business criticality.

.

SCOPE OF ACTIVITIES

The assignment will include the following actions and areas of assessment:

Stage	Tasks
Mobilisation and Preparation	• Define scope, objectives, success criteria, and assessment boundaries (apps, APIs, cloud accounts, pipelines). • Confirm maturity model, scoring rubric, evidence standards, and reporting format. • Build a sampling plan based on business criticality, exposure, and data sensitivity. • Confirm required access (repos, CI/CD, scanning tools, cloud read-only, logs, policies) and stakeholder interview plan. • Establish ways of working (workshops, cadence, risk/issue log, communications).
Application Estate Discovery and Context Building	• Inventory applications/services/APIs, owners, criticality tiers, data classifications, and key user journeys. • Identify crown jewels, trust boundaries, third parties, and major dependencies (identity, payments, messaging, data stores). • Review architecture artefacts and key non-functional requirements (availability, privacy, regulatory constraints). • Confirm top threat scenarios and abuse paths relevant to the business (account takeover, API abuse, data exfiltration).
Governance, Policy and Operating Model Assessment	• Review AppSec strategy, policies/standards, and how requirements reach engineering teams. • Assess ownership and RACI (security, engineering, platform, product) and how decisions are made. • Review risk acceptance/exception process (approvers, evidence required, expiry, review cycle). • Assess secure design review and threat modelling governance (frequency, quality, artefacts, sign-offs). • Review third-party/supplier controls as they relate to engineering delivery (libraries, SaaS, integrations)
Secure SDLC Controls and Tooling Effectiveness	• Assess SDLC controls end-to-end: requirements -> design > build > test > release > operate. • Review CI/CD security controls: branch protection, code review, gating, approvals, secrets handling, artifact promotion. • Evaluate tool coverage and signal quality: SAST, SCA, secrets scanning, IaC scanning, container scanning, DAST/API testing. • Assess integration health: PR feedback loops, riversafe.co.uk ticketing workflow, alert routing, false positive rate, tuning ownership. • Validate developer enablement: secure templates/golden pipelines, training coverage, champions programme, secure coding standards adoption.
Platform, Cloud, API and Supply Chain Deep Dives	• API security posture review: authn/authz patterns, object-level auth, schema validation, rate limiting, abuse controls, logging. • Cloud/IAM review: least privilege, workload identity, key management, segmentation, environment separation, access paths. • IaC and Kubernetes/container posture review: baseline images,

	patching, admission controls, runtime hardening, network policies, secrets management. • Software supply chain review: dependency governance, SBOM generation/storage/use, provenance/signing, registry trust, third-party exposure. • Runtime readiness review: logging standards, detection gaps, incident response playbooks for app/API threats, forensics readiness
Findings, Scoring and Risk Prioritisation	• Score maturity per domain and per sampled product/team using evidence-based criteria. • Build heatmap and identify systemic themes (root causes, not just symptoms). • Prioritise risks using criticality + exposure + exploitability + blast radius (mapped to business impact). • Identify quick wins (high impact/low effort) and foundational gaps requiring programme change. • Validate findings with stakeholders to ensure accuracy and alignment
Roadmap, Backlog and Handover	• Define target-state capabilities and a minimum security baseline (what good looks like for the organisation). • Produce a phased plan • Convert roadmap into delivery artefacts: epics/features/stories with owners, acceptance criteria, and effort bands. • Define KPIs/KRIs baseline and measurement approach (coverage, MTTR, noise rate, escape rate, gate adherence). • Run executive readout plus engineering enablement workshop; agree next steps and ownership.

SCOPE OF DELIVERABLES

The following tangible outputs will be provided as part of the engagement sign-off:

Output	Description
Executive Summary and Recommendations	• Executive-level summary of maturity baseline, top risks, and business impact. • High-level recommendations and investment priorities (what to do first and why). • Clear statement of current maturity level and target maturity objectives.
Maturity Scorecard and Heatmap	• Domain-by-domain maturity scoring with evidence rationale. • Heatmap by product/team where sampling allows (highlights inconsistency and hotspots). • Benchmark-style view showing strengths, gaps, and priority focus areas.
Detailed Technical Assessment Report	• Findings across governance, SDLC, tooling, cloud, API, supply chain, and runtime. • Evidence-backed observations with practical remediation guidance. • Identified control gaps, misconfigurations, and blind spots (process and technology).
Evidence Pack (Audit-Friendly)	• Artefacts reviewed, interviews held, repositories/pipelines sampled, tool coverage verified. • Traceability from findings to evidence to recommendation (supports audit and internal governance)
Control Coverage Matrix (SDLC x Controls)	• Mapping of SDLC stages to required controls (what exists, where, and quality rating). • Highlights missing gates, weak enforcement, and inconsistent adoption across teams.
Risk Register and Prioritised Remediation Themes	• Ranked list of key risks with impact/likelihood and affected systems. • Systemic root-cause themes (weak authz patterns, secrets hygiene, pipeline bypass risk).
30/60/90-Day Plan and Roadmap	• Sequenced plan with dependencies, milestones, and expected outcomes. • Quick wins vs foundational work vs advanced capabilities clearly separated.
Delivery Backlog Pack (Jira-Ready PBIs)	• Epics/features/stories with owners, acceptance criteria, riversafe.co.uk and effort bands. • Grouped by workstream (SDLC, API, cloud, supply chain, vulnerability management, enablement)
Target State Golden Paths Pack	• Practical templates/patterns: secure repo baseline, golden pipeline controls, API baseline, secrets pattern, logging standard. • Guidance designed to reduce friction and standardise secure delivery.
Metrics and Measurement Framework	• KPI/KRI definitions, data sources, and reporting cadence. • Recommended score tracking to prove improvement over time (not one-off assessment)
Readout Workshops and Handover	• Executive debrief and engineering enablement session to walk through findings and plan. • Confirmed owners and next-step governance to drive execution

SERVICE OUTCOMES

- Establishes a trusted maturity baseline across the full application delivery lifecycle.
- Identifies systemic AppSec gaps before they become incidents, breaches, or regulatory findings.
- Improves engineering velocity with security by focusing on enablement and golden paths, not just more gates.
- Produces a delivery-ready plan (roadmap plus backlog) that teams can execute immediately.
- Creates measurable improvement through KPIs and repeatable scoring, enabling continuous maturity uplift.