User Stories by Chapter: Application Security Program Guide

Compiled for Jordan Suber

Contents

1 Stories by Chapter **3**

2 Resequenced AppSec Program User Stories **23**

How to Use This Template

Each card maps one chapter's *Learning Goals* to a concise story, binds the chapter's *Hands-on Objectives* to concrete *Tasks*, and verifies *Outcomes* via BDD-style Acceptance Criteria. Import these cards into your backlog, tag by risk tier, and iterate.

Required Data on Every Story

- **ID** (e.g., APPSEC-1), **Title** (actionable verb), **Epic/Feature**, **Business Value** (outcome/why)
- **Priority** (Must/Should/Could), **Estimate** (SP), **Persona**, **Dependencies**, **Assumptions/Risks**
- **Acceptance Criteria** (Gherkin-ish BDD), **Tasks** (checklist), **NFR** (Security, Privacy, Reliability, etc.)

Writing Effective User Stories (Quick Guide)

Template: As a *[persona]*, I want to *[do X]* so that *[value/why]*.
INVEST: Independent, Negotiable, Valuable, Estimable, Small, Testable.
Good: "As an AppSec lead, I want a *tiered SSDLC policy* so that *teams ship securely with minimal friction*."
Anti-patterns: Vague "Research X"; multi-team mega-stories; outputs without value ("create doc") unless tied to decision/change.

1 Stories by Chapter

APPSEC-1 — Publish an AppSec Program Charter

Epic / Feature	Program Foundations
Business Value	align engineering, product, and risk on scope, value, and success criteria
Priority / Estimate	Priority: Must SP: 3
Persona	AppSec lead
Dependencies	Org strategy, security policy, product roadmap
Assumptions / Risks	Scope creep risk; time-box charter v1 and plan iterative updates

Story *As a AppSec lead, I want to Publish an AppSec Program Charter so that align engineering, product, and risk on scope, value, and success criteria.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

☐ Draft a one-page charter: mission, scope, definitions, interfaces, success metrics.

☐ Create a stakeholder map and RACI for threat modeling, testing, vuln mgmt, IR.

☐ Review with Eng/Product/Risk; capture decisions and open questions.

☐ Publish in the handbook repo; version as living document.

APPSEC-2 — Create a Control Dictionary & Traceability Matrix

Epic / Feature	Security Foundations
Business Value	give engineers clear, shared definitions and connect policies to app controls
Priority / Estimate	Priority: Must SP: 5
Persona	Security architect
Dependencies	Enterprise policies/standards
Assumptions / Risks	Terminology mismatch; include concrete code/config examples

Story *As a Security architect, I want to Create a Control Dictionary & Traceability Matrix so that give engineers clear, shared definitions and connect policies to app controls.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Compile key concepts (authn, authz, logging, crypto, secrets, input validation).
- ☐ Map each enterprise policy to concrete application controls and test evidence.
- ☐ Add links to code samples, lints, and CI checks for each control.
- ☐ Publish as `/docs/control-dictionary.md` and keep PR-able.

APPSEC-3 — Build an Application Inventory & Tiering

Epic / Feature	Program Scope
Business Value	focus effort on highest-risk apps; enable tiered controls and SLAs
Priority / Estimate	Priority: Must SP: 5
Persona	Product security engineer
Dependencies	CMDB/source of truth; service catalog
Assumptions / Risks	Owner gaps; require ownership to promote to higher envs

Story *As a Product security engineer, I want to Build an Application Inventory & Tiering so that focus effort on highest-risk apps; enable tiered controls and SLAs.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Inventory apps/services/APIs with owners, data classes, exposure, tech stack.
- ☐ Define tiering model (e.g., P0–P3) with criteria and examples.
- ☐ Record lifecycle (active/sunset), compliance drivers, and repo links.
- ☐ Export registry to CSV/JSON; integrate with CI labels per repo.

APPSEC-4 — Stand Up an App Risk Register

Epic / Feature	Risk Management
Business Value	turn threats into tracked items tied to owners, dates, and treatments
Priority / Estimate	Priority: Must · SP: 3
Persona	Risk manager
Dependencies	Inventory completed, risk rubric
Assumptions / Risks	Over-long registers stall; keep to top risks per app

Story *As a Risk manager, I want to Stand Up an App Risk Register so that turn threats into tracked items tied to owners, dates, and treatments.*

Non-Functional Performance · Security · Reliability · Accessibility · Privacy · i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Define likelihood/impact rubric and treatment options.
- ☐ Run a 60–90 min risk workshop for two critical apps.
- ☐ Create entries with owner, due date, and linkage to epics/stories.
- ☐ Establish intake workflow (new risk → triage → acceptance).

APPSEC-5 — Publish Secure Reference Architectures

Epic / Feature	Secure Design Patterns
Business Value	give teams golden paths that bake in zero-trust and least privilege
Priority / Estimate	Priority: Should SP: 5
Persona	Security architect
Dependencies	Architecture council, platform patterns
Assumptions / Risks	Architecture drift; add linters/policies to reinforce

Story *As a Security architect, I want to Publish Secure Reference Architectures so that give teams golden paths that bake in zero-trust and least privilege.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Diagram monolith, microservices, async/event-driven, and serverless patterns.
- ☐ Annotate controls per tier (authn, mTLS, input validation, logging, backups).
- ☐ Provide IaC/app templates implementing the patterns.
- ☐ Add “choose-by-facts” table and decision records (ADRs).

APPSEC-6 — Adopt a Tiered SSDLC Policy

Epic / Feature	SSDLC Alignment
Business Value	embed right-sized checks by risk tier to shift left without friction
Priority / Estimate	Priority: Must SP: 5
Persona	AppSec lead
Dependencies	Engineering buy-in, CI access
Assumptions / Risks	Over-gating; start minimal and ratchet

Story *As a AppSec lead, I want to Adopt a Tiered SSDLC Policy so that embed right-sized checks by risk tier to shift left without friction.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Define controls per SDLC phase and per tier (ASVS/SSDF-aligned).
- ☐ Wire required checks in CI (lint, SAST, SCA) with pass/fail thresholds.
- ☐ Add DoD/DoR updates to team templates referencing security checks.
- ☐ Document exceptions/waivers with expiry and approval path.

APPSEC-7 — Launch the AppSec Champions Program

Epic / Feature	Operating Model & Teams
Business Value	scale AppSec via embedded advocates and faster issue resolution
Priority / Estimate	Priority: Should SP: 3
Persona	AppSec lead
Dependencies	Managers' support, time allocation
Assumptions / Risks	Attrition/adoption risk; include incentives and community time

Story *As a AppSec lead, I want to Launch the AppSec Champions Program so that scale AppSec via embedded advocates and faster issue resolution.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Define selection rubric, responsibilities, and incentives.
- ☐ Create monthly office hours and a champions Slack channel.
- ☐ Provide starter kit (checklists, threat modeling kit, PR review guide).
- ☐ Track participation and outcomes (bugs prevented, PRs reviewed).

APPSEC-8 — Standardize Threat Modeling

Epic / Feature	Threat Modeling
Business Value	catch design flaws early and convert threats into actionable requirements
Priority / Estimate	Priority: Must SP: 5
Persona	Security champion
Dependencies	DFD notation, templates
Assumptions / Risks	Analysis paralysis; time-box sessions and prioritize

Story *As a Security champion, I want to Standardize Threat Modeling so that catch design flaws early and convert threats into actionable requirements.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Choose method (STRIDE/LINDDUN/misuse cases) and templates.
- ☐ Run two sessions on different architectures; capture DFDs and threats.
- ☐ Translate top threats into NFRs and tests.
- ☐ Add a reusable threats/mitigations catalogue to the wiki.

APPSEC-9 — Publish Secure Coding Standards

Epic / Feature	Secure Coding
Business Value	reduce recurring vulnerabilities and speed reviews with clear checklists
Priority / Estimate	Priority: Must · SP: 3
Persona	Tech lead
Dependencies	Language stacks agreed
Assumptions / Risks	One-size-fits-none risk; tailor per language

Story *As a Tech lead, I want to Publish Secure Coding Standards so that reduce recurring vulnerabilities and speed reviews with clear checklists.*

Non-Functional Performance · Security · Reliability · Accessibility · Privacy · i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Write per-language standards (input validation, encoding, secrets, crypto).
- ☐ Add PR checklists and reviewer heuristics.
- ☐ Provide pre-commit hooks and code templates.
- ☐ Run a 45-min training; record and link in the repo.

APPSEC-10 — Operationalize SAST/SCA/DAST/IAST

Epic / Feature	Security Testing
Business Value	improve signal-to-noise and make security checks part of normal CI
Priority / Estimate	Priority: Must SP: 5
Persona	Automation engineer
Dependencies	Scanner licenses, CI capacity
Assumptions / Risks	Finding overload; enforce "new high/critical = fail"

Story *As a Automation engineer, I want to Operationalize SAST/SCA/DAST/IAST so that improve signal-to-noise and make security checks part of normal CI.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Integrate SAST & SCA in CI; upload SARIF for code scanning.
- ☐ Stand up targeted DAST/IAST for a high-risk app.
- ☐ Establish severity thresholds, suppressions with expiry, and routing.
- ☐ Publish weekly trend reports and backlog hygiene metrics.

APPSEC-11 — Generate SBOMs & Sign Artifacts

Epic / Feature	Supply Chain Security
Business Value	improve provenance and compliance while enabling safe updates
Priority / Estimate	Priority: Must · SP: 5
Persona	Release engineer
Dependencies	SBOM tool, signer
Assumptions / Risks	Tooling gaps; start with top languages/images

Story *As a Release engineer, I want to Generate SBOMs & Sign Artifacts so that improve provenance and compliance while enabling safe updates.*

Non-Functional Performance · Security · Reliability · Accessibility · Privacy · i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Produce SBOM (CycloneDX/SPDX) during builds; attach to artifacts.
- ☐ Sign artifacts/images and verify in promotion gates.
- ☐ Document third-party source allowlist and review cadence.
- ☐ Add attestation checks to release workflow.

APPSEC-12 — Enforce API Security Standards

Epic / Feature	API Security
Business Value	protect data and consumers via consistent auth, validation, and quotas
Priority / Estimate	Priority: Must SP: 5
Persona	API owner
Dependencies	OpenAPI/AsyncAPI specs
Assumptions / Risks	Shadow APIs; tie standard to inventory

Story *As a API owner, I want to Enforce API Security Standards so that protect data and consumers via consistent auth, validation, and quotas.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Write API security standard (authn/z, schema validation, rate limiting).
- ☐ Add contract tests and security tests to CI.
- ☐ Gate breaking changes and insecure defaults in PRs.
- ☐ Add discovery checks for undocumented endpoints.

APPSEC-13 — Publish Cloud AppSec Baseline

Epic / Feature	Cloud-Native App Security
Business Value	set secure defaults for identity, secrets, network, and logging
Priority / Estimate	Priority: Should SP: 3
Persona	Cloud security engineer
Dependencies	Cloud org access
Assumptions / Risks	Drift risk; add config conformance packs

Story *As a Cloud security engineer, I want to Publish Cloud AppSec Baseline so that set secure defaults for identity, secrets, network, and logging.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Define shared-responsibility for app teams; list must-have controls.
- ☐ Provide bootstrap templates for logging/telemetry and secrets.
- ☐ Add guardrails and conformance checks.
- ☐ Document carve-outs and exception review.

APPSEC-14 — Harden Containers & Kubernetes

Epic / Feature	Container/K8s Security
Business Value	reduce runtime risk with minimal images and admission policies
Priority / Estimate	Priority: Must SP: 5
Persona	Platform engineer
Dependencies	Registry, admission controller
Assumptions / Risks	Breakages; start in warn mode, then enforce

Story *As a Platform engineer, I want to Harden Containers & Kubernetes so that reduce runtime risk with minimal images and admission policies.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Create minimal, scanned base images; publish usage guidance.
- ☐ Enforce image provenance and vulnerability thresholds at admission.
- ☐ Apply Pod Security standards, RBAC, and NetworkPolicies.
- ☐ Add runtime policies for sensitive syscalls and egress.

APPSEC-15 — Centralize Secrets & Workload Identity

Epic / Feature	Secrets & IAM
Business Value	eliminate hardcoded secrets and reduce blast radius via least privilege
Priority / Estimate	Priority: Must SP: 3
Persona	Service owner
Dependencies	Secrets manager, IAM
Assumptions / Risks	Migration risk; migrate one app first

Story *As a Service owner, I want to Centralize Secrets & Workload Identity so that eliminate hardcoded secrets and reduce blast radius via least privilege.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Move secrets to a managed store with rotation.
- ☐ Adopt workload identity (mTLS/JWT/OIDC) for services.
- ☐ Review and minimize IAM policies per service.
- ☐ Add secrets scanning in CI and pre-commit.

APPSEC-16 — Unify Vulnerability Intake & SLAs

Epic / Feature	Vulnerability Management
Business Value	prioritize by exploitability and asset criticality to reduce MTTR
Priority / Estimate	Priority: Must SP: 5
Persona	Vuln management owner
Dependencies	Scanner feeds, ticketing
Assumptions / Risks	Duplicate noise; dedupe by CWE/package/asset

Story *As a Vuln management owner, I want to Unify Vulnerability Intake & SLAs so that prioritize by exploitability and asset criticality to reduce MTTR.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Define prioritization (CVSS/EPSS + criticality + exposure).
- ☐ Create unified intake and dedup logic across code/deps/containers/infra.
- ☐ Set SLAs per tier and auto-create tickets with owners and due dates.
- ☐ Build dashboard (age buckets, MTTR, reopen rate).

APPSEC-17 — Integrate AppSec into Incident Response

Epic / Feature	App IR
Business Value	speed containment and comms for app-specific incidents
Priority / Estimate	Priority: Should SP: 3
Persona	IR lead
Dependencies	On-call schedule, playbooks
Assumptions / Risks	Confusion in roles; publish contact matrix

Story *As a IR lead, I want to Integrate AppSec into Incident Response so that speed containment and comms for app-specific incidents.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Write app-centric playbooks (auth bypass, data exfil, supply-chain).
- ☐ Define evidence capture and comms templates (legal/regulatory triggers).
- ☐ Run a tabletop; record actions and owners.
- ☐ Add lessons learned template and review cadence.

APPSEC-18 — Set AI/ML Security Guardrails

Epic / Feature	AI/ML Security
Business Value	prevent model abuse and data leakage with standards and tests
Priority / Estimate	Priority: Could SP: 5
Persona	ML product owner
Dependencies	Model inventory, logs
Assumptions / Risks	Novel threats; start with one model/feature

Story *As a ML product owner, I want to Set AI/ML Security Guardrails so that prevent model abuse and data leakage with standards and tests.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Threat-model one ML feature (prompt injection, data poisoning, model theft).
- ☐ Add adversarial test cases and output filters.
- ☐ Log model interactions for abuse patterns.
- ☐ Document red-team scenarios and escalation paths.

APPSEC-19 — Automate Evidence & ChatOps

Epic / Feature	Automation & Orchestration
Business Value	reduce toil and raise adoption with bots, policies-as-code, and summaries
Priority / Estimate	Priority: Should SP: 3
Persona	Automation engineer
Dependencies	Bot account, APIs
Assumptions / Risks	Alert fatigue; keep messages concise with links

Story *As a Automation engineer, I want to Automate Evidence & ChatOps so that reduce toil and raise adoption with bots, policies-as-code, and summaries.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Auto-comment PRs with scanner summaries and fix hints.
- ☐ Scaffold “new service” with secure defaults via a bot command.
- ☐ Export evidence (SBOM, test reports, approvals) automatically.
- ☐ Maintain an automation backlog with value stream mapping.

APPSEC-20 — Ship Metrics Dashboard & Maturity Plan

Epic / Feature	Metrics & Maturity
Business Value	prove risk reduction and align roadmap with measurable outcomes
Priority / Estimate	Priority: Must SP: 3
Persona	Program manager
Dependencies	Data sources, dashboard tool
Assumptions / Risks	Metric cargo-cult; define glossary and collection method

Story *As a Program manager, I want to Ship Metrics Dashboard & Maturity Plan so that prove risk reduction and align roadmap with measurable outcomes.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Choose north-star KPIs (risk reduced, MTTR, escape rate) and definitions.
- ☐ Build a dashboard with trends and targets; segment by tier/team.
- ☐ Run baseline maturity assessment (e.g., SAMM) and publish a 12-month plan.
- ☐ Review quarterly and adjust priorities based on results.

Capstone & Milestones (Reference)

Foundation: Charter, control dictionary, inventory/tiering, risk register.
Build-in Security: Reference architectures, SSDLC, champions, secure coding, testing.
Platform Guardrails: SBOM/signing, API/cloud/K8s baselines, secrets/IAM.
Operate & Improve: Vuln SLAs, App IR, AI/ML guardrails, automation, metrics+maturity.

2 Resequenced AppSec Program User Stories

Rationale for the Lifecycle Order

This sequence minimizes rework, establishes authority and budget before tooling, and pushes defect discovery as far "left" as practical while ensuring there are environments and policies to enforce decisions. It also follows common guidance from secure development frameworks (e.g., governance first, then design, then build, then test, then operate).

- **Program Foundations & Strategy**: Establish charter, scope, stakeholders, funding, and governance. Without decision rights and resourcing, downstream controls lack adoption and durability.
- **Asset & Risk Baseline**: Inventory applications, classify data, and profile threats so that standards and controls are risk-informed rather than generic.
- **Standards & Training**: Define secure coding standards, NFRs, and playbooks; stand up a

security champions network and training so teams can self-serve and build correctly the first time.

- **Design-Time Controls**: Threat modeling and architecture review catch high-severity design issues before code and infrastructure multiply the blast radius.
- **Build-Time Controls**: Integrate SAST, SCA, and secret scanning into developer workflows and CI so findings appear where developers work and before artifacts ship.
- **IaC & Cloud Security**: Apply policy-as-code to Terraform/Kubernetes and baseline cloud configurations; establish guardrails so environments match the design intent.
- **Test-Time Controls**: Exercise running builds with DAST/IAST/fuzz/API tests to find issues that static methods cannot see (auth flows, desync, runtime context).
- **Release & Runtime Controls**: Require signing/attestation and enforce deploy gates; add runtime protections (WAF, admission controllers, container runtime rules) to contain residual risk.
- **Third-Party & Supply Chain**: Govern vendors and dependencies; collect and validate SBOMs and provenance to reduce inherited risk.
- **Vulnerability Management & Pentest**: Track, triage, and remediate across the estate with clear SLAs; add periodic pentests/bug bounty to validate control efficacy.
- **Compliance & Reporting**: Map controls to standards and expose metrics/dashboards after control telemetry exists, enabling audit readiness without guesswork.
- **Program Operations & Continuous Improvement**: Stand up intake, service catalog, KPIs/KRIs, and retros to prioritize work and improve signal-to-noise over time.

Entry/Exit Guidance for Each Phase (lightweight)

- *Foundations*: Inputs: business objectives. Outputs: charter, RACI, budget, OKRs, governance cadence.
- *Asset/Risk*: Inputs: org chart, repos, cloud accounts. Outputs: app inventory, data classes, risk register.
- *Standards/Training*: Inputs: risks. Outputs: standards/NFRs, training plan, champions roster.
- *Design*: Inputs: standards. Outputs: threat models, reviewed architectures, risk tickets with owners.
- *Build*: Inputs: dev pipelines. Outputs: SAST/SCA/secret scans, SBOMs, build-hardening settings.
- *IaC/Cloud*: Inputs: IaC repos, clusters. Outputs: policy-as-code, baseline configs, drift alerts.
- *Test*: Inputs: deployable builds/env. Outputs: DAST/IAST/fuzz reports linked to backlog.
- *Release/Runtime*: Inputs: artifacts. Outputs: signed images, attestations, enforced gates, runtime rules.
- *Third-Party*: Inputs: vendors/deps. Outputs: SBOM intake, supplier risk ratings, compensating controls.

- *Vuln Mgmt/Pentest*: Inputs: findings. Outputs: SLAs, risk acceptance process, validation via tests.
- *Compliance/Reporting*: Inputs: control telemetry. Outputs: control-to-standard mappings, dashboards.
- *Ops/CI*: Inputs: all telemetry. Outputs: intake flow, service catalog, retros, maturity roadmap updates.

Quality Checks and Gaps

Use this checklist to validate coverage and identify new stories to add:

- **API Security**: Explicit stories for API design reviews, authentication/authorization patterns, and API testing (rate limits, injection, desync).
- **Secrets and KMS**: Centralized secrets management, rotation policies, and detection of hard-coded secrets across repos and images.
- **SBOM Ingestion & Policy**: Generate SBOMs in CI; ingest to a registry; enforce policies (deny vulnerable or unknown components).
- **Build Attestations & SLSA**: Sign artifacts, capture provenance, and gate releases on attestations.
- **Logging/Telemetry & Alert Routing**: Ensure applications emit security-relevant logs; define parsing, retention, and alert destinations.
- **Privacy Threat Modeling**: Add privacy misuse cases and data-minimization checks to design reviews.
- **AuthN/AuthZ Patterns**: Standardize token lifetimes, session management, and role design across services.
- **Incident Response Integration**: Connect AppSec findings to CSIRT runbooks; define rapid triage paths and severity thresholds.
- **Runtime Controls for Containers/K8s**: Admission policies, image allow-lists, runtime syscall/behavior rules.
- **Mobile/Desktop AppSec (if applicable)**: Platform-specific storage, transport, and jailbreak/root detection controls.
- **Risk-Based CD Gates**: Promotion pipelines that vary tests and approvals by risk tier and data classification.
- **Threat Intelligence Feedback**: Intake external intel and reflect in standards, tests, and block lists.
- **Metrics/OKRs**: Define KRIs and KPIs (MTTR, fix rate, mean risk score) and publish to dashboards.
- **Developer Onboarding & Self-Service**: Templates, golden repos, and paved roads that encode standards by default.
- **Title Hygiene**: Ensure story titles clearly state control area (e.g., "Threat Modeling for Service X") to improve search and automation.

Program Foundations & Strategy

APPSEC-1 — Publish an AppSec Program Charter

Epic / Feature	Program Foundations
Business Value	align engineering, product, and risk on scope, value, and success criteria
Priority / Estimate	Priority: Must SP: 3
Persona	AppSec lead
Dependencies	Org strategy, security policy, product roadmap
Assumptions / Risks	Scope creep risk; time-box charter v1 and plan iterative updates

Story *As a AppSec lead, I want to Publish an AppSec Program Charter so that align engineering, product, and risk on scope, value, and success criteria.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Draft a one-page charter: mission, scope, definitions, interfaces, success metrics.
- ☐ Create a stakeholder map and RACI for threat modeling, testing, vuln mgmt, IR.
- ☐ Review with Eng/Product/Risk; capture decisions and open questions.
- ☐ Publish in the handbook repo; version as living document.

APPSEC-6 — Adopt a Tiered SSDLC Policy

Epic / Feature	SSDLC Alignment
Business Value	embed right-sized checks by risk tier to shift left without friction
Priority / Estimate	Priority: Must SP: 5
Persona	AppSec lead
Dependencies	Engineering buy-in, CI access
Assumptions / Risks	Over-gating; start minimal and ratchet

Story *As a AppSec lead, I want to Adopt a Tiered SSDLC Policy so that embed right-sized checks by risk tier to shift left without friction.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Define controls per SDLC phase and per tier (ASVS/SSDF-aligned).
- ☐ Wire required checks in CI (lint, SAST, SCA) with pass/fail thresholds.
- ☐ Add DoD/DoR updates to team templates referencing security checks.
- ☐ Document exceptions/waivers with expiry and approval path.

APPSEC-20 — Ship Metrics Dashboard & Maturity Plan

Epic / Feature	Metrics & Maturity
Business Value	prove risk reduction and align roadmap with measurable outcomes
Priority / Estimate	Priority: Must SP: 3
Persona	Program manager
Dependencies	Data sources, dashboard tool
Assumptions / Risks	Metric cargo-cult; define glossary and collection method

Story *As a Program manager, I want to Ship Metrics Dashboard & Maturity Plan so that prove risk reduction and align roadmap with measurable outcomes.*

Non-Functional Performance Security Reliability Accessibility Privacy i18n

Acceptance Criteria (BDD)

Scenario	Happy path
Given	the target repositories, environments, and program context are available
When	the *Hands-on Objectives* for this chapter are executed
Then	the stated *Outcomes/Deliverables* for this chapter are produced, reviewed, and published

Definition of Ready: Persona clear; AC drafted; Dependencies known; Estimate set. • **Definition of Done:** All ACs pass; Tests green; Security/a11y checks; Docs updated; Deployed/flagged.

Tasks

- ☐ Choose north-star KPIs (risk reduced, MTTR, escape rate) and definitions.
- ☐ Build a dashboard with trends and targets; segment by tier/team.
- ☐ Run baseline maturity assessment (e.g., SAMM) and publish a 12-month plan.
- ☐ Review quarterly and adjust priorities based on results.

Capstone & Milestones (Reference)

Foundation: Charter, control dictionary, inventory/tiering, risk register.
Build-in Security: Reference architectures, SSDLC, champions, secure coding, testing.
Platform Guardrails: SBOM/signing, API/cloud/K8s baselines, secrets/IAM.
Operate & Improve: Vuln SLAs, App IR, AI/ML guardrails, automation, metrics+maturity.