FPT Education
FPT UNIVERSITY

Session 1

Overview of SDLC × Secure SDLC

Presenter: **Mr. Ngo Tung Son (Ph.D.)**

- Head of Information Assurance Department, FPT University, Hanoi, Vietnam
- Director of RISCS Laboratory, FPT University, Hanoi, Vietnam

FPT Education
FPT UNIVERSITY

Abbreviations & Acronyms

Abbreviation	Meaning	Explanation / Vietnamese Note
SDLC	Software Development Life Cycle	Chu trình phát triển phần mềm (Communication, Planning, Modeling, Construction, Deployment).
Secure-SDLC	Security-Integrated SDLC	Lớp bảo mật phủ (overlay) và quality gates trong toàn bộ SDLC.
CIA	Confidentiality – Integrity – Availability	Mục tiêu cốt lõi bảo mật: bảo mật, toàn vẹn, sẵn sàng.
AAA	Authentication – Authorization – Auditing	"Gold Standard" cơ chế thực thi.
AuthN / AuthZ	Authentication / Authorization	Xác thực / Ủy quyền truy cập.
SDR	Security Design Review	Cổng kiểm soát kiến trúc & bảo mật đầu vào.
Bug Bar	Bug Severity Bar / Ship Criterion	Tiêu chí chặn phát hành (C/H = 0 trước FSR).
FSR	Final Security Review	Cổng cuối: bằng chứng & rủi ro trước phát hành.
DFD	Data Flow Diagram	Sơ đồ luồng dữ liệu; đánh dấu entry points & trust boundaries.
AC-SEC-*	Acceptance Criteria – Security	Tiêu chí chấp nhận có yếu tố bảo mật trong user story/test.
SAST / DAST / IAST	Static / Dynamic / Interactive App Security Testing	Bộ kỹ thuật kiểm thử an ninh ứng dụng.
SBOM	Software Bill of Materials	Danh mục thành phần phần mềm (phục vụ phân tích CVE).
CVE	Common Vulnerabilities and Exposures	Danh mục lỗ hổng chuẩn toàn cầu.
Vault / KMS	Secret Vault / Key Management Service	Quản lý khóa & bí mật an toàn.
PEP / PDP	Policy Enforcement / Decision Point	Điểm áp thi và ra quyết định chính sách (RBAC/ABAC).
RTO / RPO	Recovery Time / Point Objective	Mục tiêu thời gian & điểm phục hồi sau sự cố.
IR	Incident Response	Quy trình phản ứng sự cố.
DFIR	Digital Forensics & Incident Response	Điều tra số & phản ứng sự cố.
PHI / PII / Fin	Protected Health / Personally Identifiable / Financial Information	Dữ liệu nhạy cảm: sức khỏe / cá nhân / tài chính.

FPT Education
FPT UNIVERSITY

Objectives

- Why Secure-SDLC must be “baked-in”, not “bolt-on”
  Vì sao Secure-SDLC phải “tích hợp sẵn”, không phải là “thêm vào”.
- Foundation: CIA (goal) ↔ AAA (implementation), “trust” & trust boundary.
  Nền tảng: CIA (mục tiêu) ↔ AAA (cách thực thi), “trust” & trust boundary.
- SDLC ↔ Secure-SDLC: 5 framework activities, security overlay, evidence store.
  SDLC ↔ Secure-SDLC: 5 hoạt động khung, overlay bảo mật, evidence store.
- Introduction to Quality Gates and Assurance.
  Threat Modeling & Quality Gates: SDR, Bug Bar, FSR.

Part 1 - Foundations & Security Basics

FPT Education
FPT UNIVERSITY

Overview of Software Security

1. CORE DEFINITION

Software Security

- Focuses on **designing**, **implementing**, and **operating** trustworthy systems.
  (Tập trung vào việc thiết kế, triển khai và vận hành các hệ thống đáng tin cậy)
- The goal is to embed security *from the beginning* of the SDLC instead of patching later.
  (Mục tiêu là đưa bảo mật vào ngay từ đầu SDLC, thay vì vá lỗi về sau)

2. OBJECTIVE

Protecting Digital Assets

- Defense throughout the lifecycle, aligned with the **CIA triad**: **Confidentiality, Integrity, Availability**.
  (Phòng vệ xuyên suốt vòng đời, gắn với bộ ba CIA: Bảo mật, Toàn vẹn, Sẵn sàng)
- Gold Standard: AuthN • AuthZ • Auditing
  (Chuẩn vàng: Xác thực • Phân quyền • Kiểm toán)

3. NATURE OF THE PROBLEM

Security Incident = Violation of Trust

- Incidents occur when **trust** is broken due to malice and incompetence .
  (Sự cố xảy ra khi niềm tin bị phá vỡ do ác ý hoặc bất cẩn)

Malice: fraud, deception, targeted attacks.
(Ác ý: gian lận, lừa đảo, tấn công có chủ đích)

Incompetence: mistakes, misunderstanding, human imperfection.
(Bất cẩn: sai sót, hiểu lầm, khiếm khuyết con người)

In practice: most exploits originate from human or process flaws → build security in early in the **SDLC**.
(Thực tế: nhiều khai thác bắt nguồn từ lỗi con người hoặc quy trình → cần đưa bảo mật vào sớm trong SDLC)

The CIA Triad & CIA as a Goal

CIA TRIAD

Confidentiality · Integrity · Availability

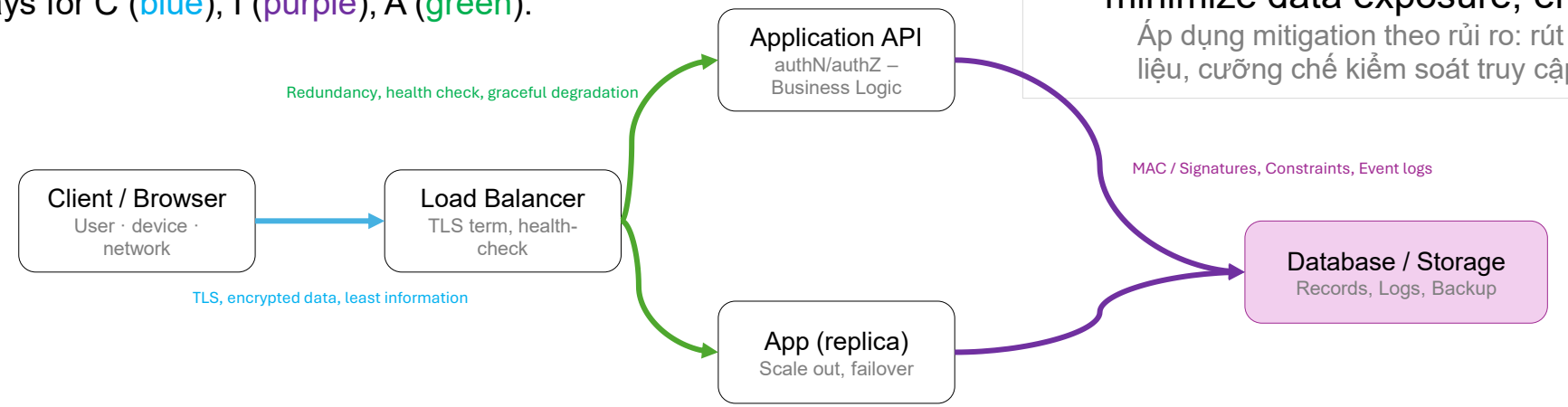
- **Confidentiality** — permit only authorized access; prevent information disclosure.
  Chỉ cho phép truy cập đã được ủy quyền; tránh rò rỉ thông tin.
- **Integrity** — keep data correct and complete; prevent unauthorized modification or deletion.
  Giữ dữ liệu chính xác & nguyên vẹn; không cho phép sửa/xóa trái phép.
- **Availability** — ensure legitimate access is timely and reliable; prevent unauthorized disruption.
  Đảm bảo truy cập hợp lệ kịp thời & tin cậy; ngăn gián đoạn trái phép.

CIA IS A GOAL

Pragmatic view of protection goals

- The CIA components are ideals — don't demand absolute perfection.
  C-I-A là các lý tưởng — không đòi hỏi sự hoàn hảo tuyệt đối.
- Minor leakages (e.g., traffic patterns) may be within acceptable risk; avoid extreme measures.
  Một số rò rỉ nhỏ (ví dụ mẫu lưu lượng) có thể nằm trong rủi ro chấp nhận được; tránh biện pháp cực đoan.
- Define "correct" via explicit authorization policy; CIA expresses goals, not mechanisms.
  Định nghĩa "đúng" bằng chính sách ủy quyền rõ ràng; CIA nêu mục tiêu, không nêu cơ chế.
- Apply risk-based mitigations: shorten exposure windows, minimize data exposure, enforce access controls.
  Áp dụng mitigation theo rủi ro: rút ngắn cửa sổ phơi lộ, tối thiểu hóa lộ dữ liệu, cưỡng chế kiểm soát truy cập.

CIA mapped over client → application/API → database, with colored overlays for C (blue), I (purple), A (green).



Principals, Credentials, Identity

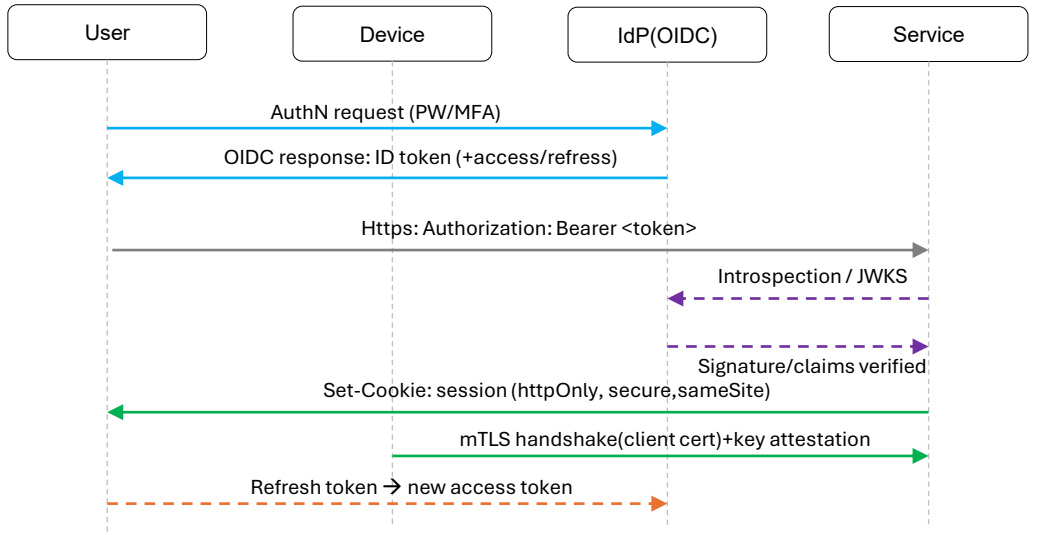
CORE DEFINITION

Principals • Credentials • Identity

- **Principal:** any actor interacting with the system: user, service, device.
  (Principal: thực thể tương tác với hệ thống: người dùng, dịch vụ, thiết bị.)
- **Credential:** proof of possession/identity (password, key, certificate, token).
  (Credential: bằng chứng sở hữu/nhận dạng như mật khẩu, khóa, chứng chỉ, token.)
- **Identity:** the authenticated identity of a principal within a specific context.
  (Identity: danh tính đã xác thực của principal trong một bối cảnh cụ thể.)

✓ **Design** must clarify *which principal acts*, which *credentials are accepted*, and *what privileges they have*.
(Thiết kế phải làm rõ ai đang hành động, chấp nhận bằng chứng nào, và đặc quyền ra sao.)

ILLUSTRATION



Description:

End-to-end: User authenticates with IdP (PW/MFA) → receives tokens → presents token to Service (RP) to establish a secure session; Device connects to Service via mTLS for machine-to-machine trust.
(Luồng tổng thể: User xác thực với IdP (PW/MFA) nhận token; đưa token cho Service để lập session an toàn; Device kết nối Service qua mTLS cho lòng tin machine-to-machine.)

Line Style Legend:

- **Solid line**: primary request/response or data flow.
  (Nét liền: luồng yêu cầu/đáp ứng chính.)
- **Dashed line**: optional/background operations, verification, or long-lived context (e.g., JWKS, refresh).
  (Nét đứt: thao tác bổ trợ/tuỳ chọn, xác minh, hoặc ngữ cảnh kéo dài.)
- **Color cues**: cyan = AuthN; gray = token to Service; violet (dashed) = introspection/JWKS; green = session/mTLS; amber (dashed) = refresh.
  (Màu sắc: xanh lam nhạt = AuthN; xám = token; tím (đứt) = introspection/JWKS; xanh lá = session/mTLS; vàng (đứt) = refresh.)

AAA Gold Standard — AuthN · AuthZ · Auditing

GOLD STANDARD

AAA are the *means* to enforce C-I-A

- **Authentication (AuthN)** — establish a trustworthy identity of the principal.
  Xác thực — xác lập danh tính đáng tin của principal.
- **Authorization (AuthZ)** — decide allow/deny by policy, based on identity and context.
  Ủy quyền — quyết định cho phép/từ chối theo chính sách dựa trên danh tính & ngữ cảnh.
- **Auditing** — tamper-resistant logs for oversight, detection, and non-repudiation.
  Kiểm toán — log chống sửa để giám sát, phát hiện và không thể chối bỏ.

AUTHENTICATION - OVERVIEW

Identity • Token-based session • Avoid partial-success leaks

Danh tính • Phiên dựa trên token • Tránh tiết lộ "thành công một phần"

- **Establish identity** with trustworthy credentials; favor **MFA** for sensitive actions.
  Xác lập danh tính bằng thông tin xác thực đáng tin; ưu tiên MFA cho hành động nhạy cảm.
- **Token-based session**: bind identity to a secure session token (cookie/header); support timeout, refresh, revoke.
  Phiên dựa trên token: ràng buộc danh tính với token phiên an toàn (cookie/header); có timeout, refresh, revoke.
- **Do not reveal partial success** on login; use rate-limits/backoff to deter guessing.
  Không tiết lộ "thành công một phần" khi đăng nhập; dùng giới hạn tần suất/trễ để chống đoán mò.

AUTHORIZATION - OVERVIEW

Allow/Deny by policy • Centralize enforcement

Quyết định allow/deny theo chính sách • Tập trung điểm cưỡng chế

- **Decisions** are based on authenticated identity, roles/attributes, and context.
  Quyết định dựa trên danh tính đã xác thực, vai trò/thuộc tính và ngữ cảnh.
- **Central guard**: enforce at a single PEP (gateway/filter); avoid scattered checks.
  Điểm canh gác tập trung: cưỡng chế tại một PEP (gateway/filter); tránh kiểm tra rải rác.
- **Use standard models**: RBAC/ABAC/PBAC; default-deny; separation of duties.
  Dùng mô hình chuẩn: RBAC/ABAC/PBAC; mặc định từ chối; tách nhiệm vụ.

AUDITING - OVERVIEW

Log critical events • Tamper-resistant • Post-incident value

Ghi sự kiện quan trọng • Chống sửa log • Vai trò sau sự cố

- **What to log**: AuthN/AuthZ events, admin actions, startups/shutdowns, updates; enough detail for *who/when/what*.
  Ghi gì: sự kiện AuthN/AuthZ, thao tác admin, khởi động/tắt, cập nhật; đủ chi tiết *ai/khi nào/làm gì*.
- **Tamper-resistant**: append-only/WORM; independent control where risk is high; monitor & alert.
  Chống sửa: chỉ-bổ-sung/WORM; quản lý độc lập khi rủi ro cao; giám sát & cảnh báo.
- **After incidents**: reconstruct timeline, determine impact; enables accountability/non-repudiation.
  Sau sự cố: tái dựng timeline, xác định ảnh hưởng; hỗ trợ trách nhiệm/không thể chối bỏ.

Trust Foundation

1. CORE CONCEPTS

Definition

- **Role of Trust**: Trust is foundational because no one builds an entire stack from scratch; we rely on external components such as hardware, the operating system, compilers, and libraries.
  (Vai trò của Niềm tin: Niềm tin là nền tảng vì không ai tự xây từ đầu; ta phải tin vào phần cứng, hệ điều hành, trình biên dịch và thư viện.)

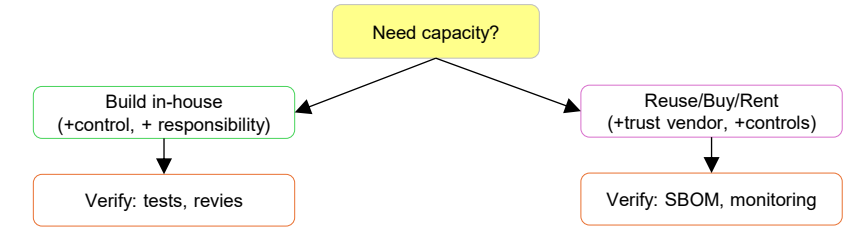
- **Trust as a Spectrum:** Trust is always granted by degree; every trust assessment carries uncertainty.
  (Niềm tin là một Phổ: Niềm tin luôn theo mức độ; mọi đánh giá đều kèm bất định.)
- **Trust Decisions:**
  - ✓ Trust too little → you must build and secure more of the system yourself.
    (Tin quá ít → bạn phải tự xây và bảo vệ nhiều phần hơn.)
  - ✓ Trust too much → higher risk of compromise or violation.
    (Tin quá mức → tăng rủi ro bị tấn công hoặc vi phạm.)

- **Implicitly Trusted Components:** Most projects depend on a technology stack (hardware, OS, tools, libraries) implicitly trusted based on vendor reputation, because verifying everything end-to-end is impractical.
  (Thành phần được tin ngầm định: dự án dựa trên ngăn xếp công nghệ được tin theo uy tín nhà cung cấp; việc tự xác minh toàn bộ là không thực tế.)

"Trust but VERIFY" Pipeline

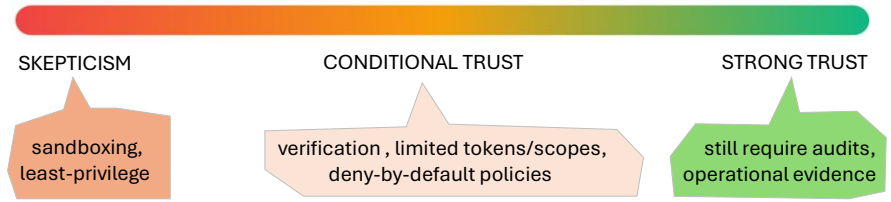
- Source/Dependencies: transparent code & dependencies, controlled provenance.
  (Nguồn/Phụ thuộc: minh bạch, kiểm soát nguồn gốc.)
- SBOM + SAST/DAST: continuous verification, early detection of vulnerabilities & license risks.
  (Xác minh liên tục, phát hiện sớm lỗ hổng & rủi ro license.)
- Signing + CD: release only verified artefacts; retain full audit trails.
  (Chỉ phát hành artefact đã xác minh; lưu dấu vết kiểm toán đầy đủ.)

2. ILLUSTRATION

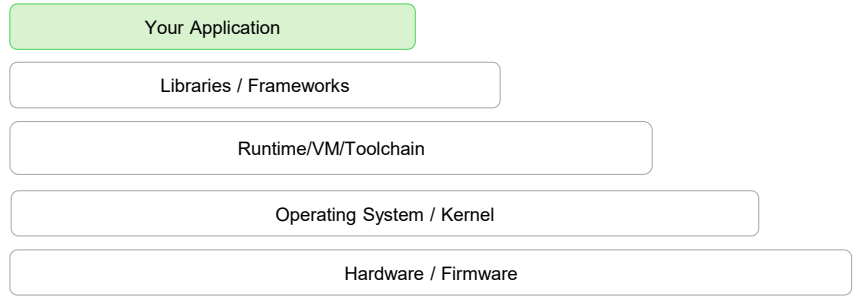


Build-vs-Buy decisions and corresponding verification layers.
(Quyết định Build vs. Buy và lớp xác minh tương ứng.)

Trust Spectrum



Implicitly Trusted Stack



Lower layers are mostly implicit trust; aim to shrink and harden the trust base.
(Phần lớn lớp dưới là trust ngầm định; cần thu nhỏ và gia cố nền tin cậy.)

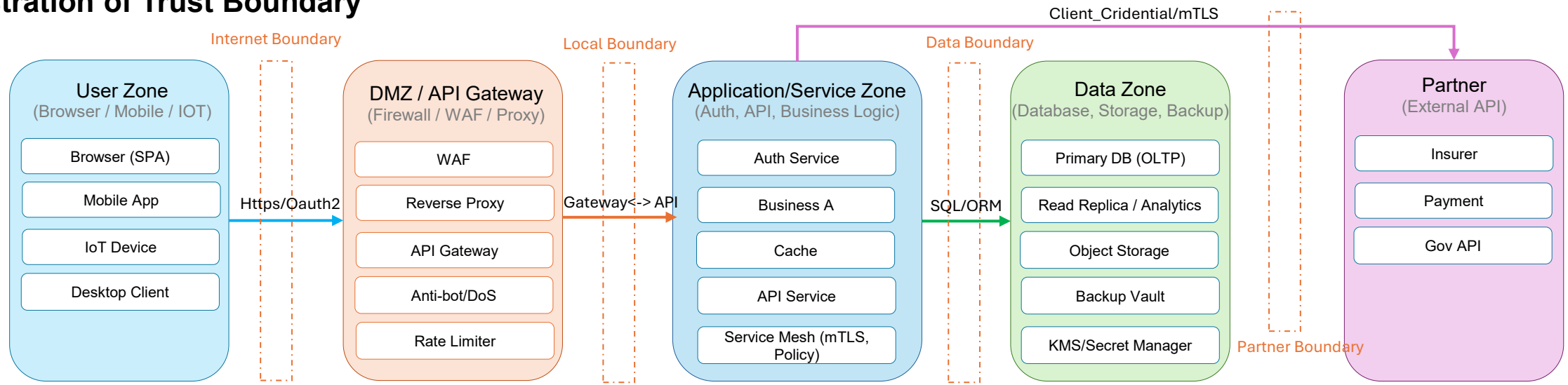
Trust Boundaries

Definition

Trust boundary is the line separating components or domains with different levels of trust. Data crossing it must be validated, authenticated, and authorized.

(Ranh giới tin cậy là đường phân tách các thành phần hoặc miền có mức độ tin cậy khác nhau. Mọi dữ liệu đi qua đều phải được xác minh, xác thực và uỷ quyền).

Illustration of Trust Boundary



Notes

- **Outer boundaries** are the most exposed; apply strict WAF, rate limits, and anomaly detection.
  (Ranh giới ngoài dễ bị tấn công nhất; cần WAF, giới hạn tần suất, phát hiện bất thường).
- **Internal boundaries** isolate subsystems and microservices with different privilege domains.
  (Ranh giới nội bộ cô lập các tiểu hệ thống và vi dịch vụ có miền đặc quyền khác nhau).
- **Data boundaries** enforce encryption-at-rest and access control per service role.
  (Ranh giới dữ liệu đảm bảo mã hoá khi lưu trữ và kiểm soát truy cập theo vai trò dịch vụ).
- **Partner boundaries** must implement API contracts, authentication, and monitoring for exfiltration.
  (Ranh giới với đối tác phải tuân thủ hợp đồng API, xác thực và giám sát rò rỉ dữ liệu).

Crystallizing the Mindset: Security-by-Design

CORE MINDSET

What is Security-By-Design?

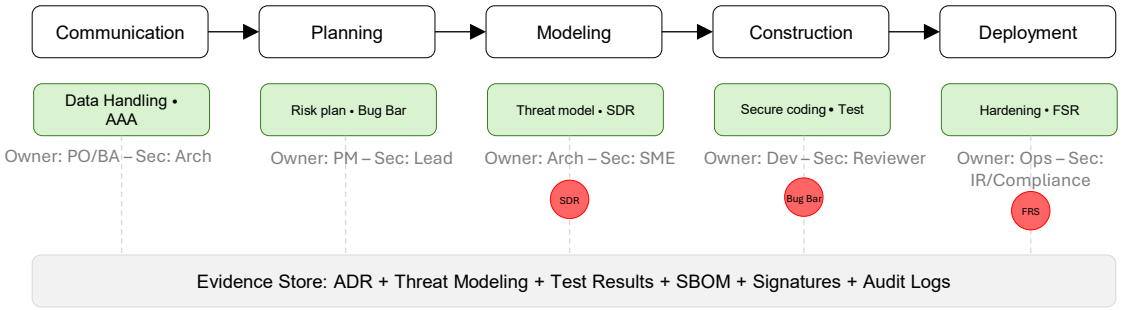
- **Definition:** Security-by-design treats security as a first-class design constraint, embedded from requirements → architecture → code & testing → deployment & operations, rather than a late-stage patch.
  (Định nghĩa: Security-by-design coi bảo mật là ràng buộc thiết kế gốc, tích hợp từ yêu cầu → kiến trúc → mã & kiểm thử → triển khai & vận hành, thay vì vá về sau.)

- **Apply the trust foundation at requirements:** identify principals, trust boundaries, sensitive data, and AAA/auditing as mandatory non-functional requirement (NFRs).
  (Áp dụng trust foundation ngay từ yêu cầu: xác định principals, trust boundaries, dữ liệu nhạy cảm, và AAA/auditing như yêu cầu phi chức năng bắt buộc.)

- **Architecture decisions must be evidenced:** each choice carries assurance evidence (threat model, tests, signatures, logs) and passes gates (SDR/Bug Bar/FSR).
  (Quyết định kiến trúc phải có bằng chứng: kèm threat model, kiểm thử, ký số, log; qua các cổng SDR/Bug Bar/FSR.)

- **Quick checklist:** up-to-date threat model; clear Bug Bar; SDR/FSR criteria; consistent AAA; 5W1H audit logs; evidence retained for audit.
  (Checklist nhanh: threat model cập nhật; Bug Bar rõ ràng; tiêu chí SDR/FSR; AAA nhất quán; log 5W1H; lưu evidence cho kiểm toán.)

Secure SDLC Overlay on the 5 Framework Activities



- **Communication**: data handling, principals & AAA early → produce *ADRs* and baseline security requirements.
  (Phân loại dữ liệu, xác định principals & AAA sớm → tạo ADR & baseline yêu cầu bảo mật.)
- **Planning:** risk plan, define Bug Bar, schedule SDR/FSR.
  (Lập risk plan, định nghĩa Bug Bar, lịch SDR/FSR.)
- **Modeling**: build a *threat model*, select controls → *SDR* before design lock.
  (Dựng threat model, chọn kiểm soát → SDR trước khi khóa thiết kế.)
- **Construction**: *secure coding*, SAST/DAST, SBOM, signing → compare against the *Bug Bar* for readiness.
  (Lập trình an toàn, SAST/DAST, SBOM, ký số → so với Bug Bar để đánh giá sẵn sàng.)
- **Deployment**: *hardening*, policy/admission, logging/monitoring → *FSR* aggregates evidence for Go/No-Go.
  (Củng cố, policy/admission, logging/monitoring → FSR tổng hợp evidence cho Go/No-Go.)

Security is an *overlay* across activities — each produces evidence for the **Evidence Store**; **gates** align with Modeling/Construction/Deployment.
(Bảo mật là lớp overlay xuyên suốt — mỗi hoạt động tạo evidence vào Evidence Store; các gate bám theo Modeling/Construction/Deployment.)

Part 2 – Introduction to Software Development Lifecycle (SDLC)

SDLC- Definition & General Structure

WHAT IS SDLC?

A Framework to transform requirements into operational software

- **SDLC** defines the structured coordination of tasks throughout the lifecycle : from *requirements → design → construction → testing → deployment & operation*.
  (SDLC là cấu trúc điều phối công việc xuyên suốt vòng đời: từ yêu cầu → thiết kế → xây dựng → kiểm thử → triển khai/vận hành.)
- It consists of **framework activities**: Communication, Planning, Modeling, Construction, and Deployment.
  (Gồm 5 hoạt động khung: Communication, Planning, Modeling, Construction, Deployment.)
- Each activity contains multiple **task sets**, which produce **work products** with defined **QA points** and **milestones**.
  (Mỗi activity chứa nhiều task set, sinh ra work product, kèm QA points và milestones.)

FRAMEWORK ACTIVITIES VS. PROFESS FLOWS

"What to do" differs from "How to proceed"

- **Framework activities**: the high-level groups of work always present in any process.
  (Framework activities: nhóm việc cấp cao, luôn tồn tại trong mọi quy trình.)
- **Process flows**: the *rhythm* or sequence *Linear* (sequential), *Iterative* (cyclic), *Evolutionary* (progressive), *Parallel* (concurrent).
  (Process flows: nhịp điệu hoặc chuỗi di chuyển — Linear, Iterative, Evolutionary, Parallel.)
- **Example:** high-risk projects prefer *iterative* to learn fast; compliance-heavy environments favor *linear/V-model* for easier audit.
  (Ví dụ: dự án rủi ro cao dùng iterative để học nhanh; môi trường tuân thủ nặng chọn linear/V-model để dễ audit.)

Summary: SDLC provides a *common language* for work, *artefacts* to verify quality, and *milestones* for managing projects.
(SDLC cung cấp ngôn ngữ chung, artefacts để kiểm chứng chất lượng, và mốc để điều hành dự án.)

CONTROL CHAIN

Transforms abstract tasks into tangible artefacts with quality checkpoints and measurable milestones — enabling traceability.
(Chuyển hóa công việc trừu tượng thành artefacts cụ thể, có điểm soát chất lượng và mốc đo lường — giúp kiểm chứng được.)

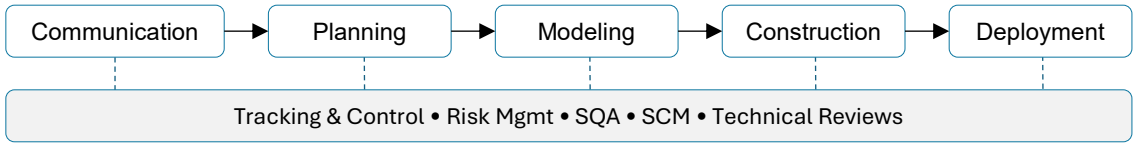
type	Task set	Work Products	Quality Assurance	Milestones
artefact	Define endpoints, data model, threats	Spec, Design, Code	Reviews, Tests	Complete, Release

Umbrella Activities across the SDLC

Cross-cutting activities that run end-to-end

Umbrella activities are recurring practices across all stages that *sustain control* of the project: tracking & control, risk management, software quality assurance (SQA), configuration management (SCM), and technical reviews (FTR/peer review). Quality comes from an *evidence-based process* at each step, not only from the final result.

Umbrella activities là các hoạt động lặp lại ở mọi giai đoạn nhằm *giữ nhịp kiểm soát* cho dự án: theo dõi & điều hành, quản trị rủi ro, đảm bảo chất lượng (SQA), quản lý cấu hình (SCM), và rà soát kỹ thuật (FTR/peer review). Chất lượng đến từ *quy trình có bằng chứng* ở từng bước, không chỉ từ kết quả cuối.



PROJECT TRACKING & CONTROL

Track work against baselines and act

- Establish **baselines** (scope, schedule, cost, quality) and measure against reality.
- At minimum: *increment schedule*, burn-up/down or EV, and an impediment list with an *owner* and a deadline.

RISK MANAGEMENT

Proactive risk governance

- Identify early, assess *probability* × *impact*, choose a strategy: avoid / mitigate / transfer / accept.
- Review on a cadence (each iteration) and tie decisions to release milestones.

SOFTWARE QUALITY ASSURANCE (SQA)

Build-in Quality, not just testing

- It's **policy + standards + procedures + QA points** woven into every activity—not only tests.
- Definition of Done by artifact; multi-layer test plans; evidence stored in an *Evidence Store*.
- Drive quality via **gates**: SDR / Bug Bar / FSR that are transparent and measurable.

CONFIGURATION MANAGEMENT (SCM)

Control versions and integrity

- Govern versions and changes of all CIs: code, docs, pipelines, IaC, configs, and release artifacts.
- Use artifact signing and SBOMs; policy/admission should allow only signed images to run in prod.
- Benefits: fewer config errors, faster recovery, better investigations.

Work Products & Traceability

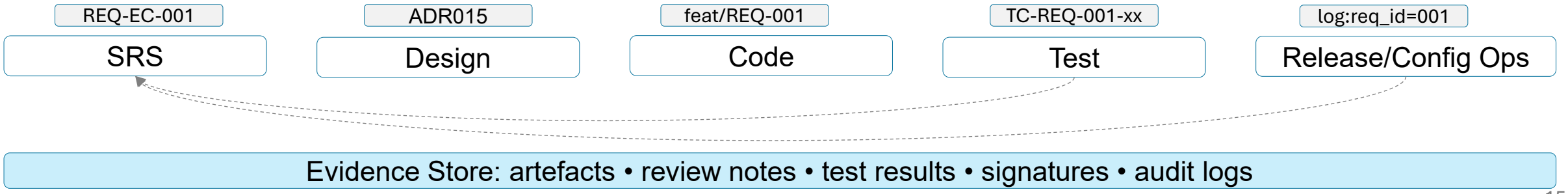
CONCEPT & GUIDING STORY

Work products = evidence carriers • Traceability = the red thread

- **Work products**: every tangible output in the SDLC (SRS, models/design, source code, test cases/reports, release config, runbooks…). They are the *evidence* to pass **QA points** and **milestones**.
  Work products: mọi kết quả hữu hình sinh ra trong SDLC (SRS, mô hình/thiết kế, mã nguồn, test case/report, cấu hình phát hành, runbook…). Chúng là *bằng chứng* để qua QA points và milestones.
- **Traceability**: recorded links between work products (e.g., *REQ → design → code → tests → release → ops log*) that support *requirements coverage*, *impact analysis*, and *auditing*.
  Traceability: các liên kết được ghi nhận giữa work products (ví dụ: *REQ → design → code → tests → release → ops log*), hỗ trợ *đảm bảo độ phủ yêu cầu*, *phân tích tác động*, *kiểm toán*.
- **Control chain**: *Task set → Work products → QA → Milestones* forms the backbone to measure quality by evidence — not gut feel.
  Chuỗi kiểm soát: *Task set → Work products → QA → Milestones* tạo xương sống để đo chất lượng dựa trên evidence — không dựa vào cảm tính.

Coverage	**Impact**	**Audit**	**Ops**
Req ↔ Test	Change ↔ Artefact	Evidence ↔ Gates	Log ↔ Usecase

ARTEFACT & TWO-WAY TRACE



Milestones & Gates

1. CONCEPT & PURPOSE

Milestone = progress checkpoint • Gate = quality gate

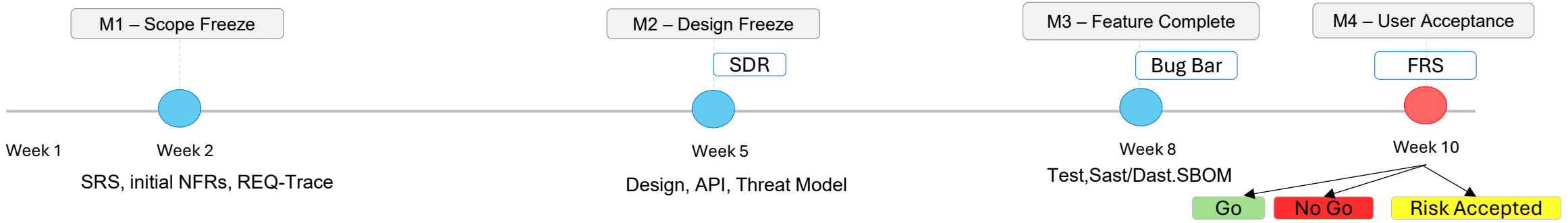
- **Milestone**: a *measurable* checkpoint that marks delivery progress (e.g., *Scope Freeze*, *Design Freeze*, *Feature Complete*).
  Milestone: mốc *đo/đếm được* để đánh dấu tiến độ (ví dụ *Scope Freeze*, *Design Freeze*, *Feature Complete*).
- **Gate**: a pass/fail *quality* checkpoint before moving forward; requires work products & QA evidence to meet criteria.
  Gate: cổng *chất lượng* kiểu pass/fail trước khi đi tiếp; cần work products & bằng chứng QA đạt tiêu chí.
- **Difference**: Milestone answers *"what is done"*; Gate answers *"good enough to proceed?"*
  Khác biệt: Milestone trả lời *"đã xong cái gì"*; Gate trả lời *"đủ tốt để đi tiếp?"*
- Tie to the chain *Task set → Work products → QA → Milestones* for transparency & verifiability.
  Gắn với chuỗi *Task set → Work products → QA → Milestones* để tăng minh bạch & khả kiểm chứng.

2. GATE CRITERIAZA B

Measurable • Observable • Auditable

- **Inputs**: required artefacts/evidence (reviewed design, updated threat model, SAST/DAST results, test reports, SBOM, signed artefacts…).
  Đầu vào: artefact/bằng chứng bắt buộc (thiết kế đã review, threat model cập nhật, SAST/DAST, báo cáo test, SBOM, artefact đã ký…).
- **Checks**: checklist/metrics (0 High/Critical; coverage ≥ X%; action items resolved).
  Tiêu chí kiểm: checklist/metrics (0 High/Critical; coverage ≥ X%; action items xử lý xong).
- **Outcome**: Go / No Go / Risk Accepted (with owner & due date).
  Kết quả: Go / No-Go / Risk Accepted (có owner & hạn xử lý).
- ✓ *Pass gate ⇒ then close the related milestone*. Evidence should be collected automatically into the Evidence Store.
  Qua gate ⇒ mới đóng mốc liên quan. Evidence nên được gom tự động vào Evidence Store.

3. TIMELINE OF MILESTONES & GATES



Standardizing Artefacts & Checklists

1. CONCEPT & GOALS

Standardize for consistency • quality control • faster onboarding

- **Artefacts** (work products): SRS, ADR/design, API specs, threat model, test plan/cases, test reports, runbooks, release config…
  Artefacts (work products): SRS, ADR/thiết kế, API spec, threat model, test plan/case, báo cáo test, runbook, cấu hình phát hành…
- **Checklists**: review/acceptance criteria for each artefact & each *gate* (SDR, Bug Bar, FSR).
  Checklist: tiêu chí review/acceptance cho từng artefact & từng *gate* (SDR, Bug Bar, FSR).
- **DoD** (Definition of Done): conditions to "close" each artefact/milestone; machine-readable if possible.
  DoD (Definition of Done): điều kiện "đóng" mỗi artefact/mốc; đọc được bởi máy càng tốt.

2. MINIMAL STARTER SET

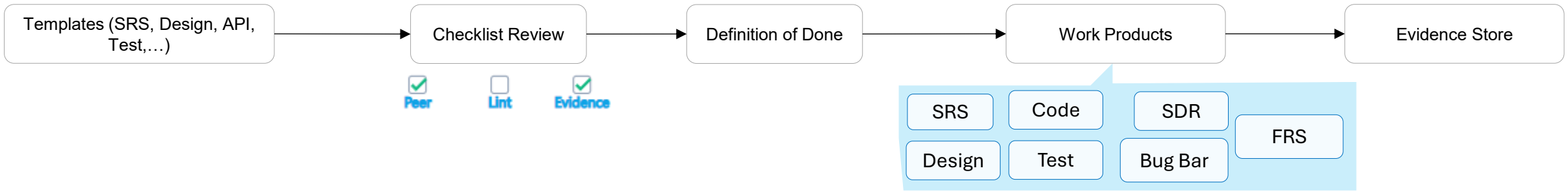
"Just-enough" templates for most projects

- **Lean SRS**: use cases, NFRs with *metric/target/workload*, REQ-* trace.
  SRS rút gọn: use case, NFR có *metric/target/workload*, trace REQ-*.
- **ADR**: context → decision → trade-offs → NFR impact → threat-model link.
  ADR: bối cảnh → quyết định → trade-offs → tác động NFR → liên kết threat model.
- **API Spec**: contract, errors, per-endpoint authZ, examples, contract tests.
  API Spec: hợp đồng, lỗi, authZ per-endpoint, ví dụ, test contract.
- **Threat Model**: assets, dataflow across *trust boundaries*, mitigations + tests.

Threat Model: tài sản, dataflow qua *trust boundaries*, biện pháp + test.

- **Test Pack**: strategy & suites (unit/integration/e2e/perf/security) + data.
  Test Pack: chiến lược & bộ test (unit/integration/e2e/perf/security) + dữ liệu.
- **Runbook & Release Notes**: config, SLOs, alerts, rollback, incident playbooks.
  Runbook & Release Note: cấu hình, SLO, cảnh báo, rollback, playbook sự cố.

3. STANDARDIZATION TURNS WORK INTO EVIDENCE

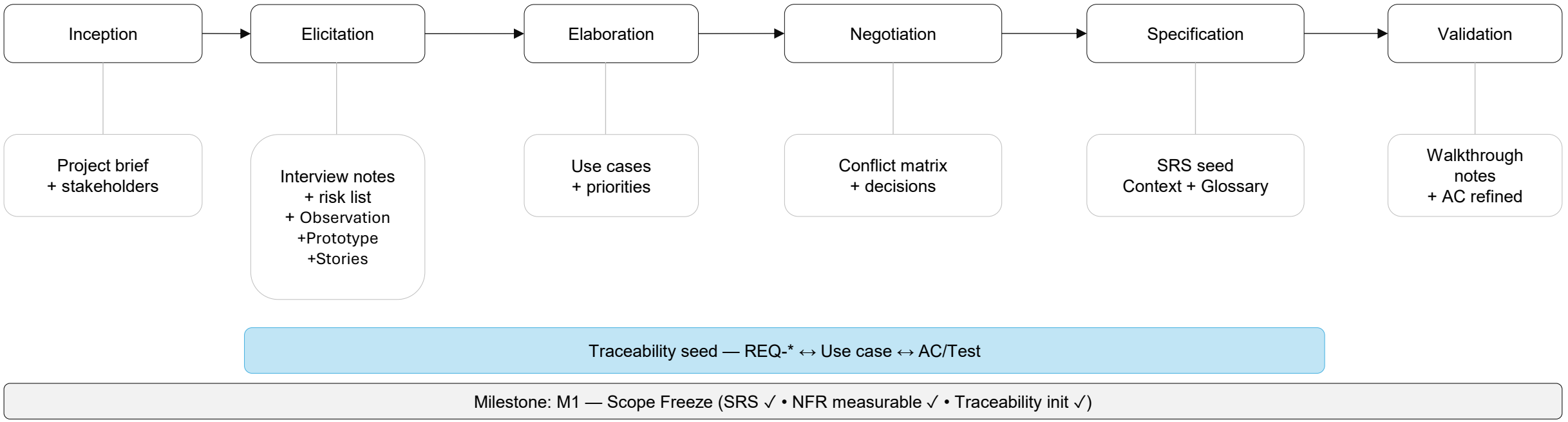


Communication — Goals & Techniques & Artefact

Turn vague ideas → testable requirements

- Action chain: **inception → elicitation → elaboration → negotiation → specification → validation**.
  Chuỗi hành động: khởi đầu → gợi ý → xây dựng → đàm phán → đặc tả → xác thực.
- Outputs: aligned scope; list of requirements & **measurable NFRs**; assumptions & constraints; *baseline artefacts* (lean SRS, context diagram, glossary).
  Đầu ra: phạm vi thống nhất; danh sách yêu cầu & NFR đo được; giả định & ràng buộc; *artefact nền* (SRS rút gọn, sơ đồ bối cảnh, từ điển thuật ngữ).
- Establish the **traceability seed** (REQ-* ↔ use case/test) for downstream reuse.
  Thiết lập traceability seed (REQ-* ↔ use case/test) để bước sau kế thừa.

Communication cycle



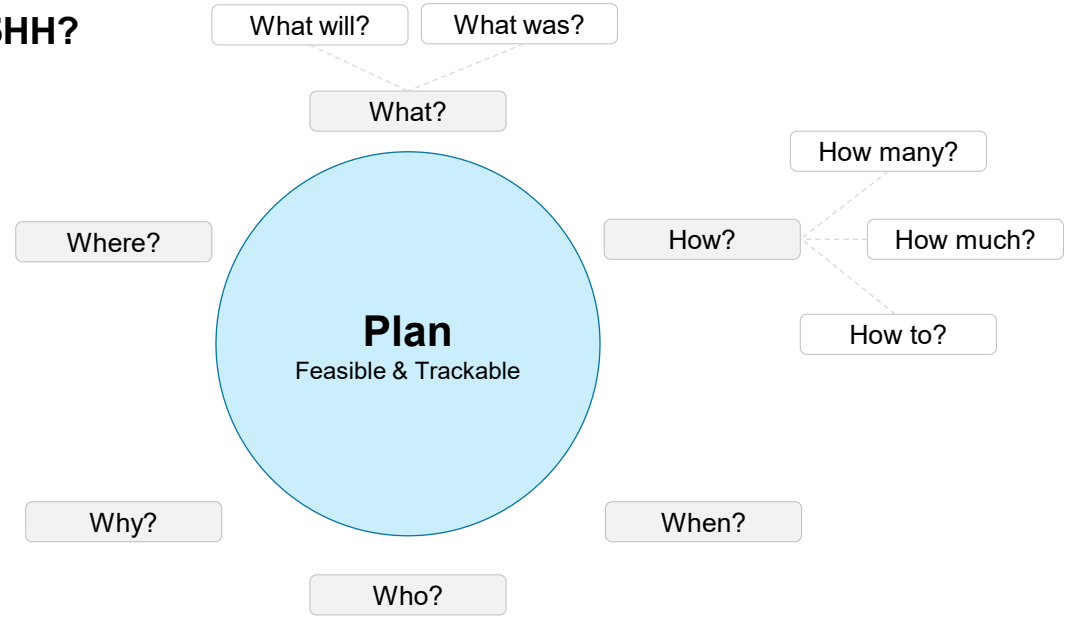
Planning – Concepts & Work Products

1. CONCEPT & GOALS

Answer the W5HH questions and produce a realistic plan

- **Scope & Feasibility**: problem statement, project constraints, success criteria.
  Phạm vi & khả thi: mô tả vấn đề, ràng buộc dự án, tiêu chí thành công.
- **Resources & Estimation**: people, environment, reusable assets; size/cost/effort estimates.
  Tài nguyên & ước lượng: nhân lực, môi trường, tài sản tái sử dụng; ước lượng kích thước/chi phí/nỗ lực.
- **Schedule & Milestones**: WBS → task network → timeline; clear deliverables.
  Lịch & mốc: WBS → mạng công việc → timeline; deliverable rõ ràng.
- **Risks & Monitoring**: identify top risks; define tracking & control mechanisms.
  Rủi ro & giám sát: nhận diện rủi ro chính; định nghĩa cơ chế theo dõi & kiểm soát.

W5HH?



2. WORK PRODUCTS

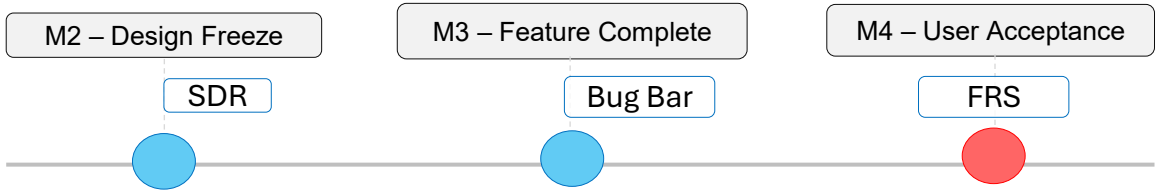
Minimum set to govern scope, resources, schedule, and risks

Work product	Purpose	Notes
Project Scope Statement Tuyên bố phạm vi	Define boundaries, objectives, constraints Xác định ranh giới, mục tiêu, ràng buộc	Problem statement; success criteria Mô tả vấn đề; tiêu chí thành công
WBS & Task Network Cấu trúc phân rã công việc	Organize work into tasks & dependencies Tổ chức công việc & phụ thuộc	Identify milestones & deliverables Nhận diện mốc & deliverable
Estimation Package Gói ước lượng	Effort/cost/schedule estimates Ước lượng nỗ lực/chi phí/lịch	Assumptions & ranges; reviewable Giả định & khoảng; có thể review
Resource Plan Kế hoạch tài nguyên	People, skills, tools, environment Nhân lực, kỹ năng, công cụ, môi trường	Acquisition & training plan Kế hoạch bổ sung/đào tạo
Risk List & Responses Danh mục rủi ro & ứng phó	Identify/assess/plan mitigations Nhận diện/đánh giá/kế hoạch giảm thiểu	Top-N risks with owners & triggers Top-N rủi ro, người phụ trách & ngưỡng
Master Schedule (Gantt) Lịch tổng thể	Timeline with checkpoints & slack Timeline có checkpoint & buffer	Baseline + change control rules Baseline + quy tắc thay đổi

3.PLAN FOR SECURITY

Security-by-Plan: tie security to milestones, budget, and DoR/DoD

- Include security in **scope / WBS / estimation / schedule**; iterate per *increment*.
  Đưa bảo mật vào scope / WBS / ước lượng / lịch; lặp theo từng *increment*.
- Pass gates with **evidence**: documents, test reports, configurations.
  Qua các cổng bằng bằng chứng: tài liệu, báo cáo kiểm thử, cấu hình.
- Gate lineup: **SDR → Bug Bar → FSR**.
  Chuỗi cổng: SDR → Bug Bar → FSR.



Modeling – Requirements & Architecture & Data

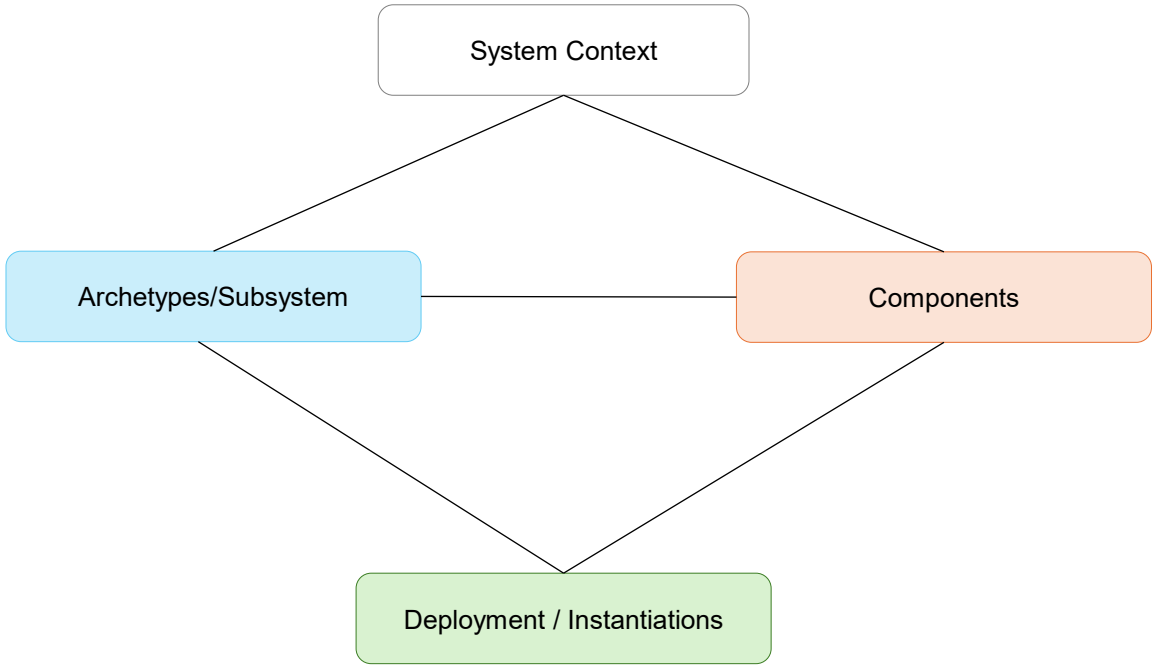
1. DESIGN MODEL

Design elements : Requirements • Data • Architecture • Interface • Component • Deployment

- **Requirements Modeling** are first-class in the design model.
  RM là ưu tiên hang đầu trong mô hình thiết kế
  - ✓ **Functional (FR)**: behaviors/services observable by actors; captured by scenarios, use cases, flows, states.
    Chức năng (FR): hành vi/dịch vụ quan sát được bởi tác nhân; thể hiện qua kịch bản, use case, luồng, trạng thái.
  - ✓ **Non-functional (NFR)**: quality attributes (performance, security, availability, observability…); modeled with measurable scenarios & budgets.
    Phi chức năng (NFR): thuộc tính chất lượng (hiệu năng, bảo mật, khả dụng, quan sát…); mô hình bằng kịch bản đo được & ngân sách.
- **Data design** — structures and organization derived from analysis.
  Dữ liệu — cấu trúc & tổ chức xuất phát từ phân tích.
- **Architectural** — overall structure and relations among parts.
  Kiến trúc — cấu trúc tổng thể & quan hệ giữa các phần.
- **Interface** — how parts interact (UI, APIs, external/internal interfaces).
  Giao diện — cách các phần tương tác (UI, API, giao diện ngoài/nội bộ).
- **Component-level** — implementation detail at module/class level.
  Thành phần — chi tiết cài đặt mức mô-đun/lớp.
- **Deployment** — mapping software onto its runtime environment.
  Triển khai — ánh xạ phần mềm lên môi trường chạy.

2. ARCHITECTURE VIEWS

Three core views: Context • Components • Deployment



Flow: context → characterize (archetypes) → refine to components → describe deployment.

Construction – Coding & Testing

1. OBJECTIVE & SCOPE

Turn design into verifiable software

- **Coding standards** and **peer review** as the foundation for quality.
  Coding standard & peer review làm nền cho chất lượng.
- **Unit → Integration → System → Acceptance (UAT)** per increment; run a *regression suite* continuously to preserve already-correct behavior.
  Unit → Integration → System → Acceptance (UAT) theo increment; *Regression suite* chạy xuyên suốt để bảo toàn hành vi đã đúng.
- Tie work to **QA points** and **gates** (Unit/Integration/System/UAT) before release decisions.
  Gắn với QA points & các gates (Unit/Integration/System/UAT) trước quyết định phát hành.
- Aggregate evidence (tests/reports/configs) into the Evidence Store to support Go/No-Go.
  Evidence (test/report/config) gom về Evidence Store để quyết định Go/No-Go.

2. CODING

Standardize and enforce discipline when coding

- Consistent naming/layout/comments; split into cohesive *modules*.
  Naming/layout/comment nhất quán; tách *modules* cohesive.
- Comply with interface/API contracts from design.
  Tuân thủ interface/API contract từ thiết kế.
- Branches/commits trace to requirements (e.g., feat/REQ-123) to maintain *traceability*.
  Nhánh/commit bám yêu cầu (ví dụ feat/REQ-123) để giữ *traceability*.

3. TESTING

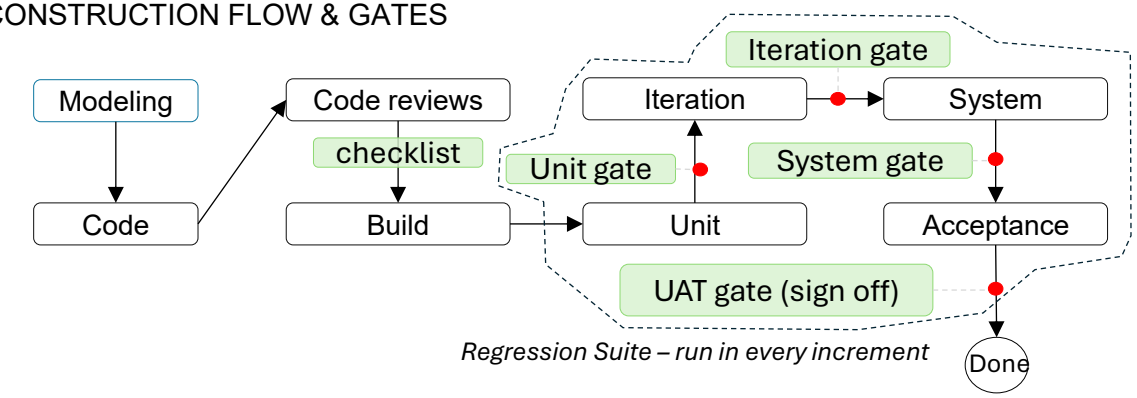
Test scope & regression control

Levels: Unit → Integration → System → Acceptance (UAT).

- **Unit:** verify small logic with dependencies isolated; fast feedback.
  Unit: xác minh logic nhỏ, cô lập phụ thuộc; phản hồi nhanh.
- ***Integration***: verify contracts/flows across modules/services.
  Integration: xác minh hợp đồng/luồng giữa modules/services.
- **System:** end-to-end on a production-like environment; touch API /DB /queues/ 3rd-party.
  System: kiểm *end-to-end* trên môi trường gần production; chạm API/DB/queue/3rd-party.
- **Acceptance:** confirm business AC via user journeys; prepare sign-off.
  Acceptance: xác nhận *business AC* theo user journeys; chuẩn bị sign-off.

Evidence: test reports, logs/traces, environment configs; store in the **Evidence Store** to support decisions at the gate.
Bằng chứng : báo cáo test, log/traces, cấu hình môi trường; lưu về **Evidence Store** để phục vụ decision tại gate.

4. CONSTRUCTION FLOW & GATES

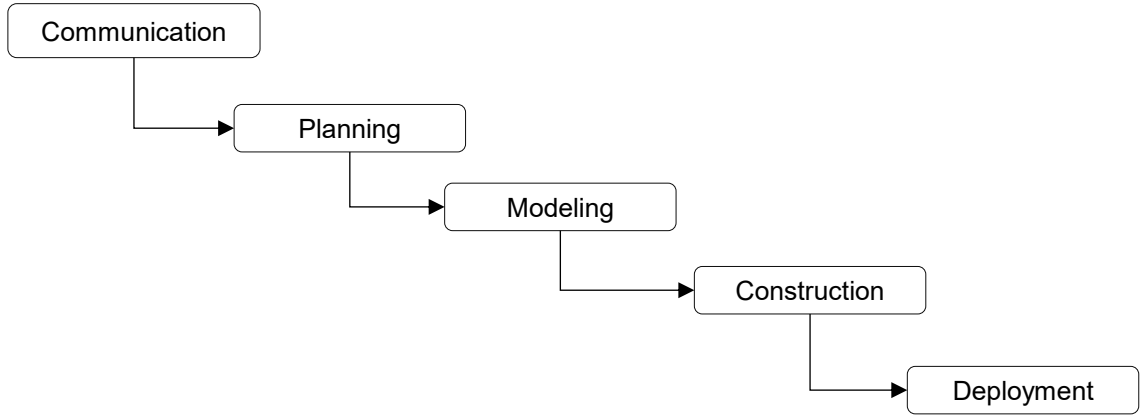


Linear models: Waterfall & V-Model

1. WATERFALL

Linear sequence • rigorous docs • high cost of change

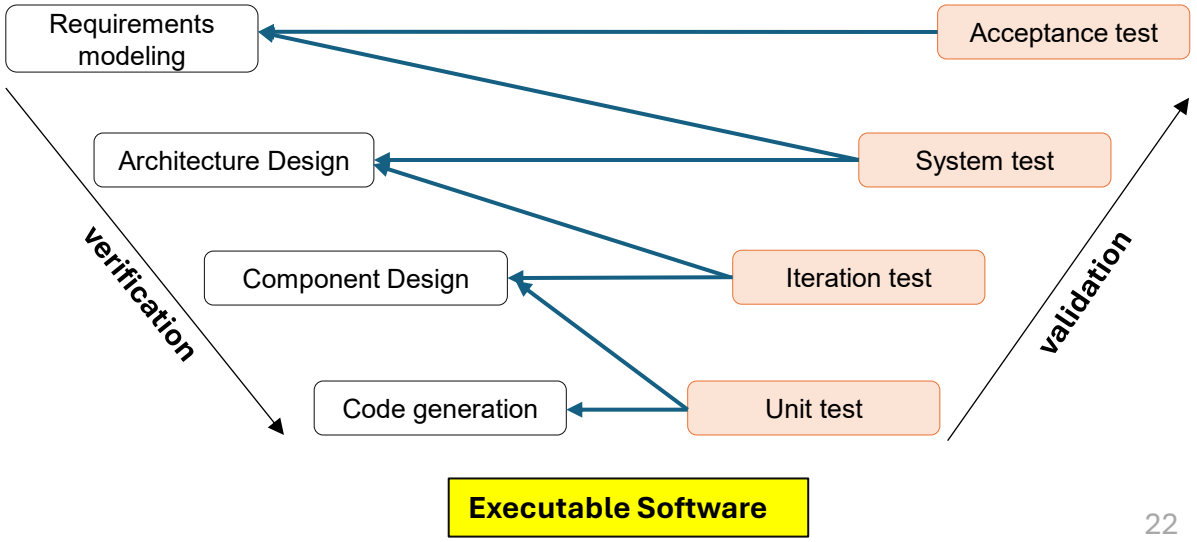
- Best when **requirements are stable** and scope is clear; heavy *compliance* demands strict **audit/documentation**.
  Phù hợp khi yêu cầu ổn định, phạm vi rõ; bối cảnh *tuân thủ nặng*, cần audit/tài liệu chặt.
- **Process flow** Communication → Planning → Modeling → Construction → Deployment; easy to place **gates** and control progress.
  Luồng Communication → Planning → Modeling → Construction → Deployment; dễ đặt **gates** & kiểm soát.
- **Cons**: running product appears *late*; changes introduce *blocks* & high costs; drift risk if requirements lack clarity.
  Nhược: sản phẩm chạy được xuất hiện *muộn*; thay đổi gây *block* & chi phí cao; dễ lệch nếu yêu cầu thiếu rõ.



2. V-MODEL: DEVELOPMENT AND VERIFICATION/VALIDATION

Each development phase maps to a verification / validation phase

- Visualizes **Verification/Validation** across technical phases; strengthens **traceability**.
  Nhìn hoá mối quan hệ Verification/Validation với các pha kỹ thuật; tăng traceability.
- Pairings: *Requirements* ↔ *Acceptance*, *Architectural* ↔ *System*, *Component* ↔ *Integration*, *Code* ↔ *Unit*.
- Useful in healthcare and regulated fields: substantial evidence; test plans are prepared *early* from the specs.
  Hữu ích trong lĩnh vực y tế/quy định chặt: cần chứng cứ dày; test plan được chuẩn bị *sớm* theo đặc tả.

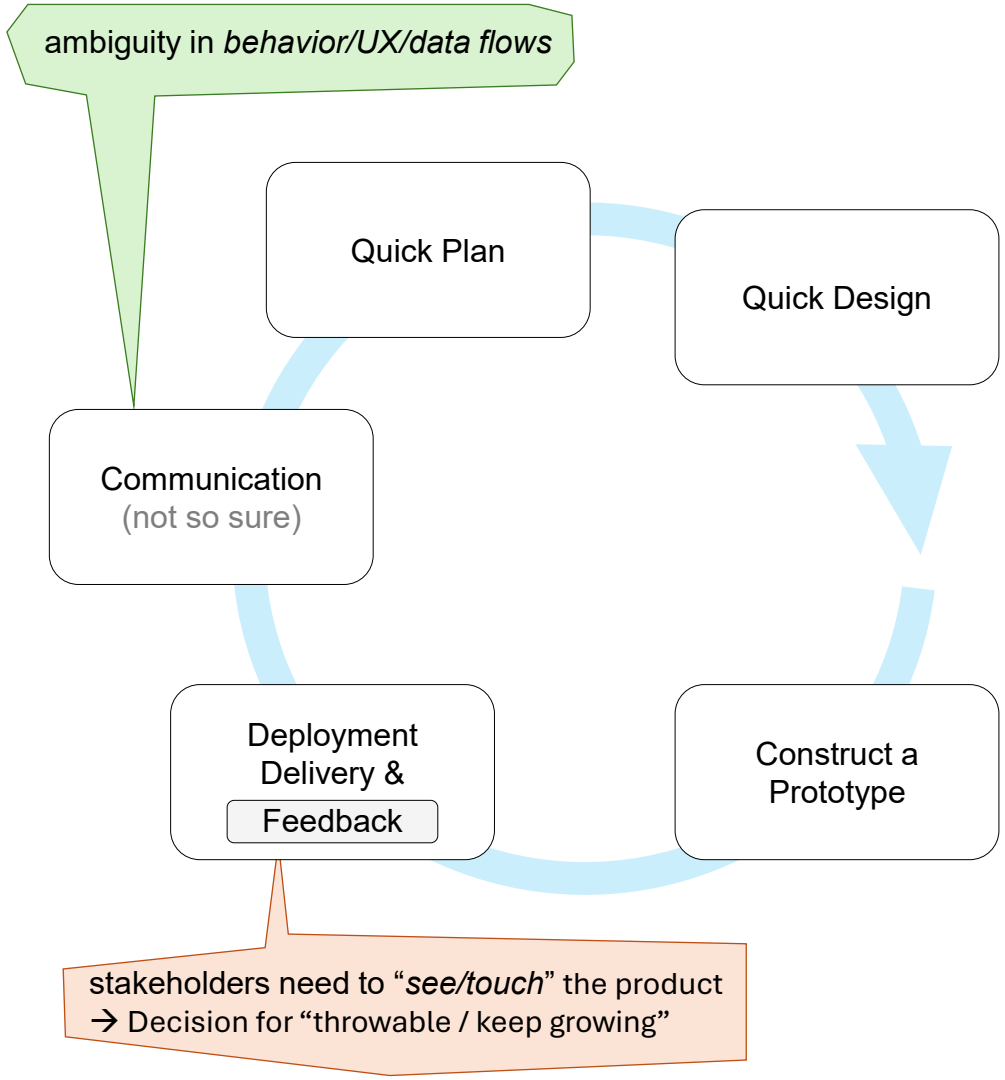


Prototyping – Reducing Requirements Risk

OVERVIEW

Clarify expectations before making big investments

- **Goal**: use a *prototype* to **discover/validate requirements**, reduce misunderstandings, align expectations.
  Mục tiêu: dùng *prototype* để khám phá/xác nhận yêu cầu, giảm hiểu lầm, đồng bộ kỳ vọng.
- **Condensed flow**: Communication → *Quick plan* → *Quick design* → *Build prototype* → *Evaluate with users* → *Refine requirements* → (iterate) → **Specification baseline** → Engineering build.
  Quy trình (rút gọn): Communication → *Quick plan* → *Quick design* → *Build prototype* → *Evaluate với người dùng* → *Refine requirements* → (lặp) → Specification baseline → Engineering build.
- **Two flavors**: *Throwaway* (discard after learning) and *Evolutionar* (grow into the product); both require appropriate quality discipline.
  Hai cách: *Throwaway* (vứt sau khi học) và *Evolutionary* (nâng dần thành sản phẩm); cách nào cũng cần kỷ luật chất lượng.
- **Pros:**
  - ✓ **Unclear requirements**—ambiguity in *behavior/UX/data flows*.
    Yêu cầu chưa rõ—mơ hồ về *hành vi/UX/luồng dữ liệu*.
  - ✓ **Highly interactive UI** or broad scope: stakeholders need to "*see/touch*" it.
    Giao diện giàu tương tác hoặc phạm vi rộng: cần được "*trông thấy/chạm vào*".
  - ✓ **Expectation contract**: users review/comment; team updates SRS/design with traceability.
    Hợp đồng kỳ vọng: người dùng duyệt/góp ý; nhóm cập nhật SRS/design có truy vết.
  - ✓ **Cost avoidance later**: find mismatches early to avoid expensive rework.
    Tiết kiệm chi phí muộn: phát hiện sai lệch sớm, tránh sửa lớn khi đã build.



Spiral — Risk Driven Lifecycle

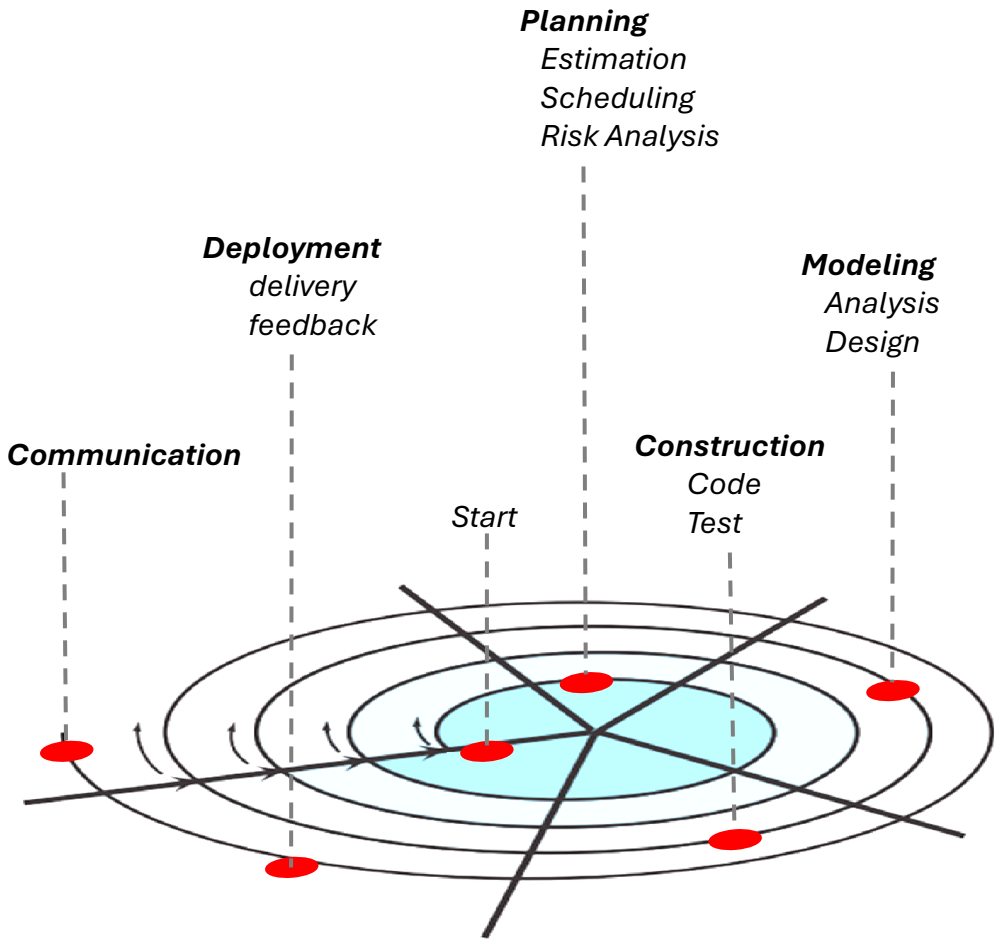
OVERVIEW

Risk-driven model with anchor-point milestones

- **Risk-driven**: each cycle emphasizes *identifying & treating risks*; the product evolves across releases (model/prototype → fuller versions).
  mỗi vòng lặp nhấn mạnh *xác định & xử lý rủi ro*; sản phẩm tiến hoá qua các bản phát hành (model/prototype → phiên bản hoàn thiện hơn).
- **Cyclic growth**: degree of *definition* and *realization* increases while *risk decreases*; **anchor-point milestones** secure stakeholder commitment.
  mức *xác định* & *hiện thực* tăng dần trong khi *rủi ro giảm*; các mốc anchor-point bảo đảm cam kết stakeholders.
- **Across the lifecycle**: concept → new product dev → enhancement; each cycle revises plan, cost, and schedule.
  từ concept → phát triển sản phẩm mới → mở rộng; mỗi vòng điều chỉnh kế hoạch, chi phí, lịch.

When to use: High uncertainty, new tech, significant risks

✓ Unclear requirements/solutions with many options and assumptions.
  Yêu cầu/giải pháp chưa rõ với nhiều lựa chọn và giả định.
✓ Hard-to-control dependencies (vendors, APIs, infra).
  Phụ thuộc khó kiểm soát (nhà cung cấp, API, hạ tầng).
✓ Need evidence at commitment milestones before scaling.
  Cần bằng chứng tại các mốc cam kết trước khi mở rộng.



Concurrent - State-Driven Development

OVERVIEW

Process network: activities exist concurrently in observable states

- **Concurrent development model** (a.k.a. concurrent engineering): captures the *parallel & iterative* nature of work across SDLC models.
  thể hiện phần *song song & lặp* trong các mô hình SDLC.
- Each *activity/action/task* (e.g., **Communication/Modeling/Construction**) is always in a visible **state** (*Inactive, Under development, Under review, Baselined, Under revision, Awaiting changes, Done*).
  Mỗi *activity/action/task* (ví dụ Communication/Modeling/Construction) luôn nằm trong một trạng thái quan sát được (*Inactive, Under development, Under review, Baselined, Under revision, Awaiting changes, Done*).
- **Events** in one area can *trigger* state changes elsewhere (e.g., "analysis model correction" pulls requirements from *done* back to *awaiting changes*).
  Events ở một điểm có thể *kích hoạt* chuyển trạng thái ở điểm khác (ví dụ "analysis model correction" kéo requirements từ *done* về *awaiting changes*).

WHEN TO USE

Many teams/streams in parallel, need a real-time state view

- ✓ Fits **product engineering** with multiple teams; reflects the actual project state instead of forcing linearity.
  Phù hợp product engineering nhiều nhóm; phản ánh trạng thái thực của dự án thay vì ép tuyến tính.
- ✓ **Dependencies visible** via explicit events and state transitions ⇒ fewer conflicts, better coordination.
  Phụ thuộc được minh hoạ qua sự kiện & chuyển trạng thái ⇒ giảm xung đột, phối hợp tốt.
- ✓ Helps optimize *time-to-market* while keeping control via states and *gates*.
  Giúp tối ưu *time-to-market* nhưng vẫn kiểm soát bằng trạng thái & *gates*.

Part 3 – Secure Overlay inside SDLC

Secure SDLC — Layered Thinking & Principles

PRINCIPLES

Defense-in-depth on top of a disciplined SDLC

- **Policy & governance**: risk management, security policy, asset classification.
  Chính sách & quản trị: quản trị rủi ro, chính sách an ninh, phân loại tài sản.
- Least privilege & separation of concerns; *fail-secure*, secure defaults.
  Đặc quyền tối thiểu & tách biệt mối quan tâm; *fail-secure*, cấu hình mặc định an toàn.
- Complete mediation, auditability, tamper-evident logs.
  Kiểm soát toàn diện, khả năng kiểm toán, log chống sửa.
- **Economy of mechanism**: simpler secure designs, reviewed & threat-modeled.
  Tối giản cơ chế: thiết kế an toàn gọn, đã review & threat-model.

CORE SECURITY CONCEPTS

CIA goals • Attack surface • Misuse cases

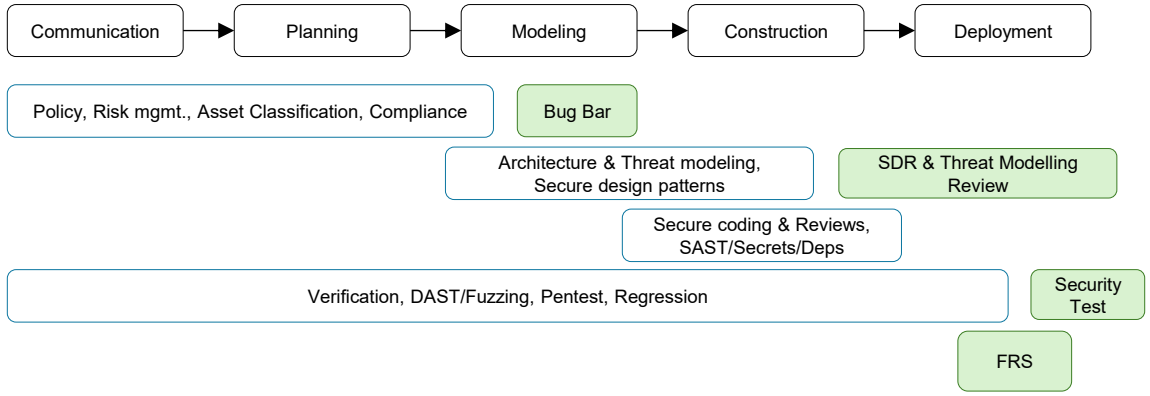
- **CIA goals**: protect *Confidentiality*, *Integrity*, *Availability* with layered controls.
  bảo vệ *bí mật*, *toàn vẹn*, *sẵn sàng* bằng các lớp kiểm soát.
- **Attack surface**: all entry/exit points (UI, API, data flows, admin); reduce via *least privilege*, validated inputs/outputs, simple designs.
  mọi điểm vào/ra (UI, API, luồng dữ liệu, admin); giảm bằng *đặc quyền tối thiểu*, kiểm tra vào/ra, thiết kế đơn giản.
- **Misuse/abuse cases**: negative scenarios to complement user stories; drive AC-SEC-* and regression tests.
  kịch bản tiêu cực bổ sung user story; dẫn dắt AC-SEC-* và test hồi quy

OVERLAY ACROSS SDLC ACTIVITIES

Inject security & set gates

- **Communication/Requirements**: security requirements, *misuse/abuse cases*, data classification.
  Giao tiếp/Yêu cầu: yêu cầu an ninh, *misuse/abuse cases*, phân loại dữ liệu.
- **Planning**: risk register, acceptance criteria with security (AC-SEC-*).
  Lập kế hoạch: sổ rủi ro, tiêu chí chấp nhận có bảo mật (AC-SEC-*).
- **Modeling**: architecture review, *threat modeling*, security design patterns.
  Modeling: duyệt kiến trúc, *threat modeling*, mẫu thiết kế an toàn.
- **Construction**: secure coding standards, SAST/secret scan, dependency hygiene.
  Lập trình: chuẩn mã an toàn, SAST/scan secret, quản lý phụ thuộc.
- **Testing**: DAST/Fuzzing, security regression, penetration test.
  Kiểm thử: DAST/Fuzzing, hồi quy an ninh, pen-test.
- **Deployment/Operations**: hardening, keys/secrets, monitoring/IR/patching.
  Triển khai/Vận hành: hardening, khóa/secret, giám sát/ứng phó vá lỗi.

LAYERED SECURITY & GATES

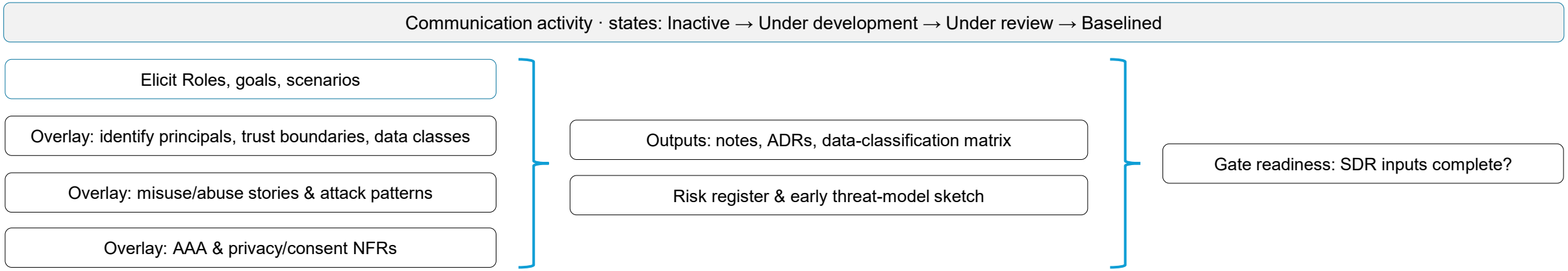


Secure SDLC — Security x Communication Overlay

SECURITY OVERLAY INSIDE COMMUNICATION

Make security a first-class topic in elicitation

- **Stakeholders & trust boundaries**: map actors (users, services, devices), data flows, and trust zones.
  lập bản đồ tác nhân, luồng dữ liệu, vùng tin cậy.
- **Security NFRs**: CIA goals; AAA (AuthN/AuthZ/Auditing); privacy/consent; resilience.
  mục tiêu CIA; AAA; quyền riêng tư/đồng ý; khả năng chịu đựng.
- **Misuse/abuse & attack patterns**: elicit negative scenarios; reuse known patterns.
  khai thác tình huống tiêu cực; tái sử dụng mẫu phổ biến.
- **Data classification** & handling rules (collection, storage, sharing, retention).
  Phân loại dữ liệu & quy tắc xử lý (thu thập, lưu trữ, chia sẻ, lưu giữ).
- **Traceability**: tag acceptance criteria AC-SEC-* and link to use/misuse cases.
  Khả năng truy vết: gắn tiêu chí chấp nhận AC-SEC-* và liên kết tới use/misuse case.
- **Evidence** for gates: meeting notes, ADRs, risk log, early threat-model sketch → feed SDR later.
  Bằng chứng cho các cổng: biên bản, ADR, sổ rủi ro, phác thảo threat-model → phục vụ SDR.



Non-Functional Requirements (NFR)

CONCEPT & ROLE

NFRs describe how the system operates & is perceived

- **NFR = quality attributes**: performance, **security**, availability, observability/monitoring, compliance, scalability…
  NFR = thuộc tính chất lượng: hiệu năng, bảo mật, khả dụng, quan sát/giám sát, tuân thủ, mở rộng…
- **Embedded in the process**: weave NFRs into artefacts, QA points, and milestones → evidence via work products.
  Không đứng ngoài quy trình: dệt NFR vào artefact, QA points & milestones → có bằng chứng qua work products.
- **Mindset**: make them *writable • measurable • testable* — not slogans.
  Tư duy: hãy viết được • đo được • kiểm thử được — không phải khẩu hiệu.

3. EXAMPLES OF NFRS

- **Performance**: percentile targets under realistic load; e.g., read patient record ≤ 300 ms @p95, 200 RPS.
  Hiệu năng: mục tiêu theo phân vị dưới tải thực; VD: đọc hồ sơ ≤ 300 ms @p95, 200 RPS;
- **Security**: *100% sensitive endpoints accept only valid identities; session/credentials expire ≤ 60 minutes; default-deny + role-based grants; 100% critical actions are **audited**.*
  Bảo mật: *100% endpoint nhạy cảm chỉ chấp nhận yêu cầu có danh tính hợp lệ; phiên/credential hết hạn ≤ 60 phút; mặc định deny-by-default và cấp theo vai trò; 100% hành động quan trọng đều được audit*
- **Observability**: logs/metrics/traces with correlation IDs; audit event = identity+time+action+resource+outcome; 95% audit queries ≤ 5 s.
  Quan sát: logs/metrics/traces có correlation ID; sự kiện audit = identity+time+action+resource+outcome; 95% truy vấn audit ≤ 5 s.

HOW TO WRITE GOOD NFRS

- Clear title + template: When [condition/workload], the system must [goal + metric] within [constraints/resources]. ”
  Tiêu đề rõ + khuôn mẫu: Khi [điều kiện/workload], hệ thống phải [mục tiêu + metric] trong [ràng buộc/tài nguyên].
- Design: record why the target is achievable (architecture, constraints, performance budget).
  Thiết kế: nêu vì sao mục tiêu đạt được (kiến trúc, ràng buộc, ngân sách hiệu năng).
- Code/Config: implement decisions (DB indexes, caches, connection limits, flags…).
  Code/Cấu hình: hiện thực quyết định (index DB, cache, giới hạn kết nối, flags…).
- Test/Monitor: turn targets into tests + dashboards; put reports/signatures into the Evidence Store.
  Test/Monitor: biến mục tiêu thành bài test + dashboard; đưa báo cáo/chữ ký vào Evidence Store.
- Backward trace: when tests/alerts fire, trace back to the NFR & ADR to adjust.
  Backward trace: khi test/cảnh báo kích hoạt, truy ngược về NFR & ADR để điều chỉnh.

- **Availability**: SLO 99.9%, RTO 15′, RPO 5′; multi-AZ, auto-failover; incident runbooks tested quarterly.
  Khả dụng: SLO 99.9%, RTO 15′, RPO 5′; multi-AZ, auto-failover; runbook sự cố kiểm thử hàng quý.
- **Compliance**: 180-day retention for audit logs; test-env data masking; consent logging; periodic audits.
  Tuân thủ: retention 180 ngày cho audit log; masking dữ liệu ở môi trường test; lưu consent; kiểm toán định kỳ.

Secure SDLC — Plan x Security Overlay

PRINCIPLES

Security-by-Plan: tie security to milestones, budget, DoR/DoD

- Include security in **scope/WBS/estimates/schedule**; iterate by *increment*.
  Đưa security vào scope/WBS/ước lượng/lịch; lặp theo *increment*.

Pass gates with **evidence**: docs, test reports, configs.
Qua cổng bằng bằng chứng: tài liệu, báo cáo test, cấu hình.

Gates: **SDR** → **Bug Bar** → **FSR**.

BUDGET & STAFFING

Fund the minimum-security work items

- **WBS**: threat modeling, SDR packet, SAST/DAST/secret scan, SBOM, hardening, chaos/negative tests.
  WBS: threat modeling, hồ sơ SDR, SAST/DAST/secret scan, SBOM, hardening, test tiêu cực/chaos.
- **Cost of quality (security)**: buffer for review → fix → re-test; periodic scans in CI/CD; IaC policy-as-code.
  Chi phí chất lượng (an ninh): đệm thời gian review → sửa → re-test; scan định kỳ trong CI/CD; IaC policy-as-code.
- **Owners**: SDR packet, threat-model upkeep, triage tool findings, independent gate approver.
  Chủ sở hữu: hồ sơ SDR, cập nhật threat-model, phân loại kết quả tool, người duyệt cổng độc lập.

DOR & DOD

Fund the minimum-security work items

DEFINITION OF READY

✓ Use case & **data classification** (PII/PHI?).
Use case & phân loại dữ liệu (PII/PHI?).

✓ Acceptance criteria cover *abuse/misuse cases*.
Tiêu chí chấp nhận bao phủ *abuse/misuse cases*.

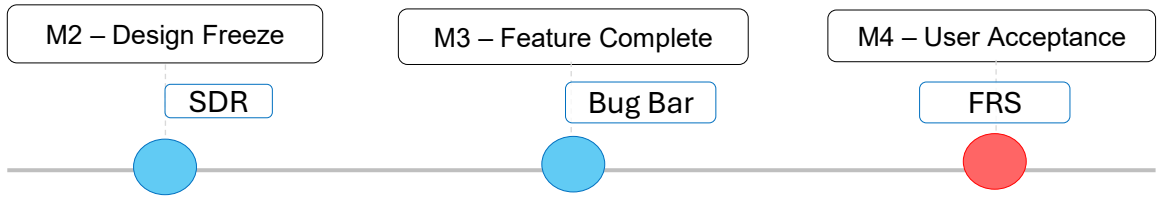
✓ New flows/data → update **threat model** draft & test plan.
Luồng/dữ liệu mới → cập nhật bản nháp threat model & kế hoạch test.

DEFINITION OF DONE

✓ Role-based AuthZ; structured logs without PII leaks.
Phân quyền theo vai trò; log có cấu trúc, không lộ PII.

✓ SAST/DAST/secret scans: **Critical/High = 0**; dependencies have **SBOM** & policy pass.

✓ Evidence stored in **Evidence Store** + tracker links.



Secure SDLC – Security x Modeling Overlay

1. MODELING GOALS

From analysis to design models that expose risk & control points

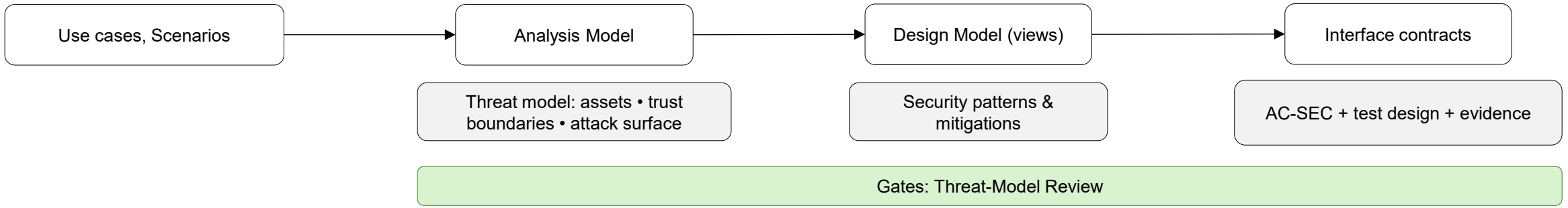
- **Select views**: context • containers • components • deployment; define interfaces/contracts & quality attributes.
  Chọn góc nhìn: bối cảnh • thùng chứa • thành phần • triển khai; xác định giao diện/hợp đồng & thuộc tính chất lượng.
- **Map data & flows**: classify data, draw trust boundaries, list entry/exit points
  phân loại dữ liệu, vẽ biên tin cậy, liệt kê điểm vào/ra.
- **Traceability**: tie scenarios to acceptance criteria & tests.
  Truy vết: gắn kịch bản với tiêu chí chấp nhận & kiểm thử.

2. SECURITY OVERLAY INSIDE MODELING

Threat modeling • Patterns • Trade-offs

- **Identify** assets, entry points, trust boundaries; sketch DFDs; relate misuse/abuse cases and attack patterns.
  Nhận diện tài sản, điểm vào, biên tin cậy; phác DFD; liên hệ misuse/abuse case.
- **Assess risks** across CIA & traceability (audit/logging); consider dependencies & suppliers.
  Đánh giá rủi ro theo CIA & truy vết (audit/log); xét phụ thuộc & nhà cung cấp.
- **Choose patterns**: AuthN/AuthZ (RBAC/ABAC), input validation, output encoding, encryption (in transit/at rest), rate-limit, secure session, key/secret handling.
  Chọn mẫu: xác thực/phân quyền (RBAC/ABAC), kiểm tra đầu vào, mã hoá đầu ra, mã hoá đường truyền/lưu trữ, hạn tốc, phiên an toàn, quản lý khoá/secret.
- **Derive AC-SEC** and test design; define evidence for later gates (SDR/Bug Bar/FSR).
  Suy ra AC-SEC & thiết kế test; xác định bằng chứng cho các cổng sau (SDR/Bug Bar/FSR).

3. MODELING ARTEFACTS WITH SECURITY OVERLAY



Secure SDLC – Security x Constructions (Code + Test) Overlay

1. CODING OVERLAY

From secure standards to verifiable evidence

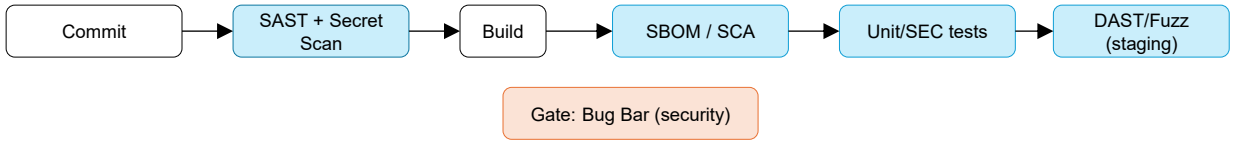
- **Secure coding standards** (input validation, output encoding, error handling, secure session).
  (kiểm tra đầu vào, mã hoá đầu ra, xử lý lỗi, phiên an toàn).
- **Secrets & keys**: no hard-coded secrets; use Vault/KMS; rotate; least privilege.
  Khoá & bí mật: không hard-code; dùng Vault/KMS; xoay định kỳ; đặc quyền tối thiểu.
- **Dependencies**: SBOM + policy; update strategy; known-vuln triage with owners.
  Phụ thuộc: SBOM + policy; chiến lược cập nhật; phân loại lỗ hổng với người phụ trách.
- **Code review** with security checklist; pair review for risky modules; link evidence to PRs/ADRs.
  Review mã với checklist an ninh; review đôi cho module rủi ro; gắn bằng chứng vào PR/ADR.
- **Traceability**: implement AC-SEC-*; structured logs/audit hooks.
  Truy vết: hiện thực AC-SEC-*; log có cấu trúc/audit hook.

2. SECURITY OVERLAY INSIDE TESTING

Negative, adversarial & evidence-driven testing

- **Unit & component tests**: authorization matrix; input validation; crypto config sanity.
  Unit & component test: ma trận phân quyền; kiểm tra đầu vào; cấu hình mã hoá.
- **Security regression**: test cases from fixed vulns & misuse stories.
  Hồi quy bảo mật: test rút từ lỗi đã vá & misuse stories.
- **Dynamic testing**: DAST against staging; fuzz parsers; canary endpoints.
  Kiểm thử động: DAST trên staging; fuzz các bộ phân tích; canary endpoints.
- **Pen-test readiness**: scope & rules of engagement; evidence capture.
  Sẵn sàng pen-test: phạm vi & quy tắc; thu thập bằng chứng.

SECURE CI/CD WITH GATES



CODE AREAS → CONTROLS/TESTS

Code Area	Controls	Tests	Evidence
API endpoints	AuthN/AuthZ – Rate limit – validation	Unit + negative – DAST	Logs, test reports, ADR
Data Access Layer	Record- Level ACL - Encryption	AuthZ tests – Performance with crypto	Audit trail, benchmark log
Build / Release	SBOM – Signing – Policy	Dependencies Scan, supply-chain test	SBOM, signatures, attestation

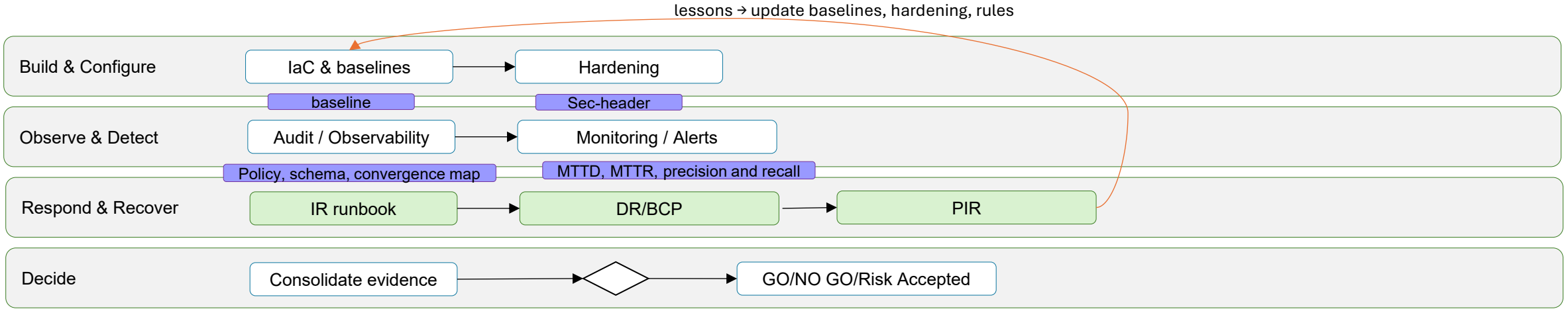
Secure SDLC: Deployment and Operations

OBJECTIVE

Evidence-driven ops: ready to investigate & recover

- **Hardening**: secure-by-default, least privilege, complete mediation.
  Hardening: mặc định an toàn, đặc quyền tối thiểu, kiểm soát toàn diện.
- **Audit & observability**: append-only, tamper-evident logs for AuthN/AuthZ, admin, config.
  Ghi vết & quan sát: log chỉ-thêm, chống sửa cho đăng nhập/ủy quyền, thao tác quản trị, cấu hình.
- **Monitoring/alerts**: risk-based rules; measure alert precision/recall and MTTR.
  Giám sát/cảnh báo: quy tắc theo rủi ro; đo độ chính xác/bao phủ cảnh báo và MTTR.
- **IR/DR**: runbook Detect → Contain → Eradicate → Recover → Learn; regular DR drills.
  IR/DR: sổ tay Phát hiện → Cô lập → Khử → Phục hồi → Rút kinh nghiệm; diễn tập DR định kỳ.
- **FSR pre-Go-Live**: weigh evidence → Go/No-Go/Risk-accepted.
  FSR trước Go-Live: dựa trên bằng chứng → Go/No-Go/Chấp nhận rủi ro.

OPS STACK & FSR GATE



Software Design Review – Goals + Timing + Dossier

1. SDR GOALS

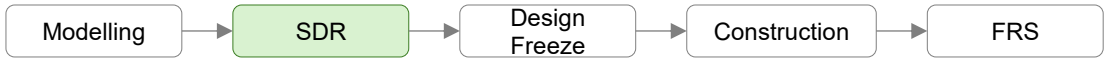
Confirm critical threat controls before *design freeze*

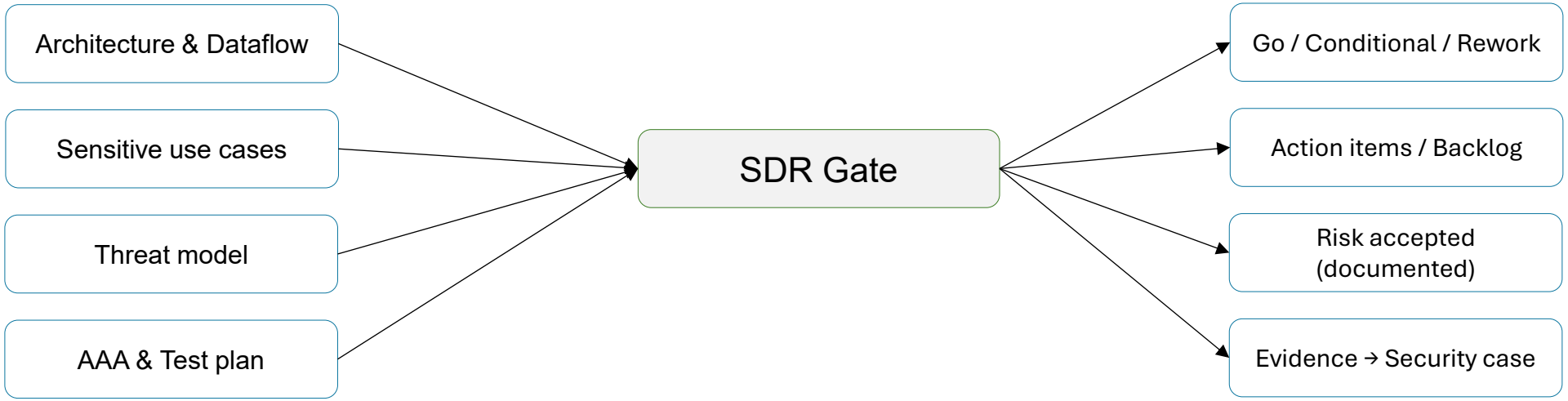
- **Top risks** from threat modeling have **mitigations** in the right architectural locations.
  Top rủi ro từ threat modeling đã có biện pháp đặt *đúng chỗ* trong kiến trúc.
- **AAA at system level**: deny-by-default, least privilege, complete mediation.
  AAA cấp hệ thống: mặc định từ chối, đặc quyền tối thiểu, kiểm soát toàn diện.
- **Gaps triaged** into *action items* with owners and due dates.
  Khoảng trống được phân loại thành *action items* có người phụ trách & hạn xử lý.
- **Evidence** consolidated (architecture diagrams, risk table, test criteria) into the *security case*.
  Bằng chứng gom lại (sơ đồ kiến trúc, bảng rủi ro, tiêu chí test) vào *security case*.

2. TIMING & FACILITATION

Right before *design freeze*; repeat on architecture change

- After Modeling: have architecture views, dataflows, threat modeling (four questions), AAA plan.
  Sau Modeling: đã có các view kiến trúc, dataflow, threat modeling (bốn câu hỏi), kế hoạch AAA.
- Before major build commitments to reduce cost of change.
  Trước khi cam kết xây dựng lớn để giảm chi phí thay đổi.
- Invite: architect, security lead, dev lead, QA, ops rep, PO/PM.
  Thành phần: kiến trúc sư, security lead, dev lead, QA, đại diện ops, PO/PM (tài liệu đọc trước ≤ 10 trang).





FPT Education
FPT UNIVERSITY

Bug Bar — Severity Thresholds & Scope

1. OBJECTIVE

Set security-defect thresholds at gates to control risk

- **Clear definitions** of *severity* & mandatory actions before each milestone M1–M4.
  Định nghĩa rõ ràng mức *severity* & hành động bắt buộc trước từng mốc M1–M4.
- **Consistency** across dev, QA, security, ops with one decision standard.
  Nhất quán giữa dev, QA, security, ops theo một chuẩn ra quyết định.
- **Tied to QA points** : failing the Bug Bar ⇒ fail the gate.
  Gắn với điểm QA : không đạt Bug Bar ⇒ không qua cổng.

2. SCOPE OF APPLICATION

Apply uniformly across four layers

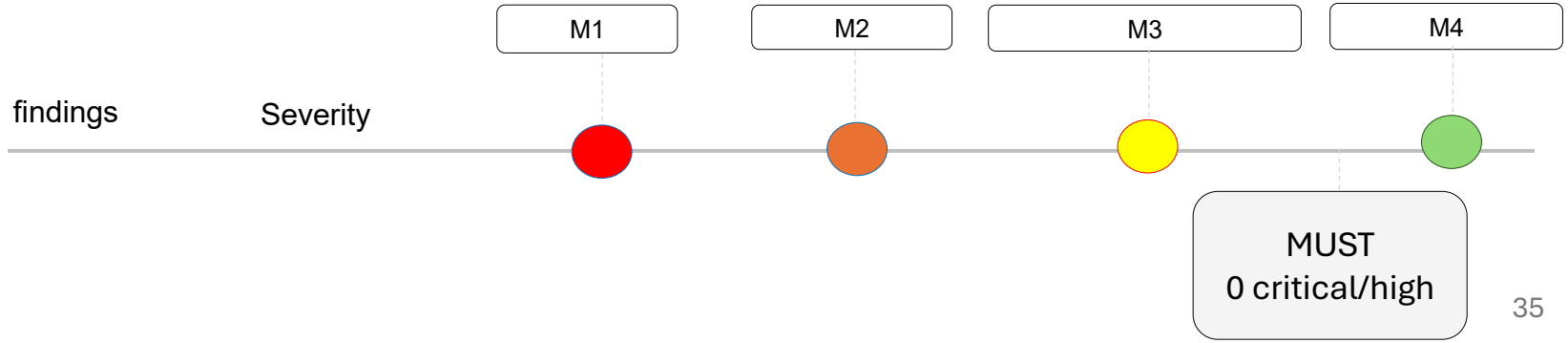
- **Code**: logic/security bugs (injection, AAA bypass, session handling).
  Mã: lỗi logic/bảo mật (injection, lách AAA, xử lý phiên).
- **Configuration**: unnecessary ports/services, weak TLS, unsafe defaults.
  Cấu hình: cổng/dịch vụ thừa, TLS yếu, mặc định không an toàn.
- **Dependencies:** vulnerable versions (CVE), untrusted sources.
  Phụ thuộc: phiên bản có CVE, nguồn không tin cậy.
- **Pipeline**: insufficient controls (signing, policy-as-code), missing SAST/DAST/IAST/secret scans.
  Pipeline: thiếu kiểm soát (ký, policy-as-code), thiếu quét SAST/DAST/IAST/secret.

3. SUGGESTED SEVERITY CLASSES (ILLUSTRATIVE)

Concise definitions to align teams

- Critical : leads to broad unauthorized PHI access/write; RCE; breaks core AAA.
  gây truy cập/ghi PHI trái phép diện rộng; RCE; phá vỡ AAA cốt lõi.
- Hight : privilege escalation; exploitable injection; leaked production secrets.
  leo thang đặc quyền; injection khai thác được; lộ secret môi trường sản xuất.
- Medium misconfig reducing security posture but with temporary mitigations.
  Medium: sai cấu hình làm giảm an toàn nhưng có biện pháp tạm thời.
- Low **:** hardening/recommendations with low impact; does not block release.
  Low: khuyến nghị/hardening, ảnh hưởng thấp; không chặn phát hành.

- Suggested milestones: M1 (end of Modeling) • M2 (start of Construction) • M3 (System test) • M4 (FSR/Pre-release)
- Critical/High ⇒ 0 before M4; Medium ⇒ mitigation & fix plan; Low ⇒ assigned owner



FSR — Final Checks & Ship/No-Ship Decision

1. OBJECTIVE

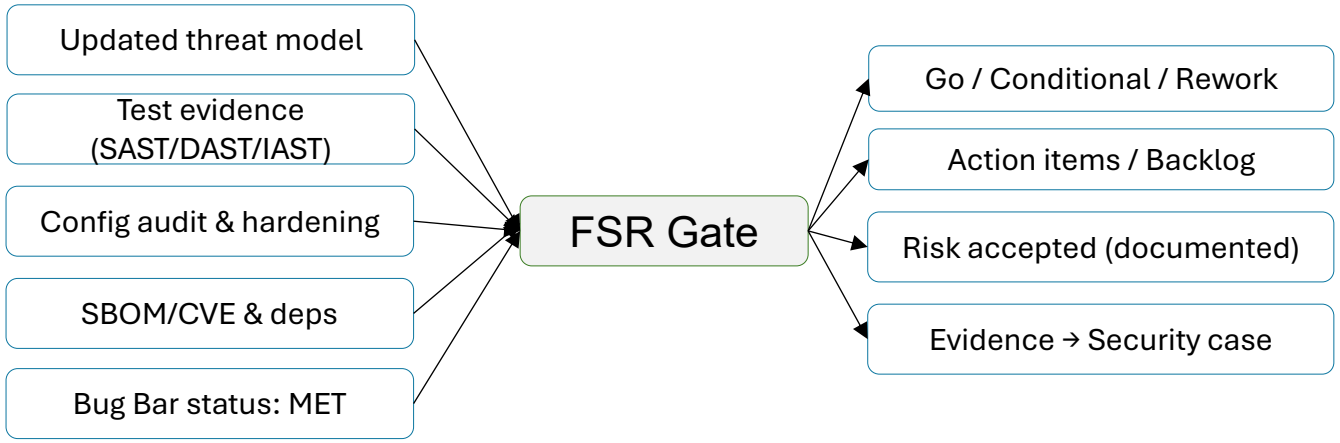
Assure sufficient evidence to *ship / no-ship*

Final verification that security controls run as designed.
Rà soát cuối xác nhận các kiểm soát hoạt động đúng thiết kế.

Security case is complete: claim → argument → evidence.
Hồ sơ an ninh đủ: mệnh đề → lập luận → bằng chứng.

Align with **Bug Bar** & QA criteria: fail bar ⇒ fail the gate.
Đối chiếu Bug Bar & tiêu chí QA: không đạt bar ⇒ không qua cổng.

Lock **post-release monitoring** & a scheduled PIR.
Chốt giám sát hậu phát hành & mốc PIR.



2. INPUT & GATE CRITERIA

What must exist before FSR

Updated threat model with top risks, mitigations, and residual assumptions.
Threat model cập nhật với rủi ro chính, biện pháp và giả định còn lại.

Test evidence: SAST/DAST/IAST; security test cases (expected ↔ actual) with pass/fail.
Bằng chứng kiểm thử: SAST/DAST/IAST; test bảo mật (kỳ vọng ↔ thực tế) có kết quả pass/fail.

Config audit: baselines/hardening; AAA enabled & verified; retrievable audit samples.
Audit cấu hình: baseline/hardening; AAA bật & đã kiểm chứng; có mẫu log truy xuất.

Dependencies: SBOM + CVE report; 0 Critical/High unresolved or justified exceptions.
Phụ thuộc: SBOM + báo cáo CVE; 0 Critical/High chưa xử lý hoặc loại trừ có biện giải.

Bug Bar met: C/H = 0; M mitigated & scheduled; L has owner.
Bug Bar đạt: C/H = 0; M có biện pháp & lịch; L có người phụ trách.

3. DECISION & RISK ACCEPTANCE

How to record in the security/assurance case

- **Ship**: all criteria met; post-release ops prepared.
  Phát hành: mọi tiêu chí đạt; vận hành hậu phát hành đã chuẩn bị.
- **Conditional ship**: residual risk justified; monitoring conditions set.
  Phát hành có điều kiện: rủi ro còn lại có biện giải; điều kiện giám sát rõ ràng.
- **No-Ship**: fails Bug Bar/QA or lacks required evidence.
  Không phát hành: không đạt Bug Bar/QA hoặc thiếu bằng chứng bắt buộc.
- **Risk accepted**: record scope, rationale, review date, accountable owner.
  Chấp nhận rủi ro: ghi rõ phạm vi, lý do, ngày xem xét lại, người chịu trách nhiệm.

Summary

Summary – Overview Secure-SDLC

SUMMARY

Secure-SDLC = overlay layer on SDLC to establish quality gates (SDR, Bug Bar, FSR) and collect evidence throughout the lifecycle.

- Trust → CIA (Confidentiality-Integrity-Availability) is the goal; Gold Standard (AAA) — Authentication, Authorization, Auditing — is the enforcement mechanism.
  Trust → CIA (Confidentiality-Integrity-Availability) là mục tiêu; Gold Standard (AAA) — Authentication, Authorization, Auditing — là cơ chế thực thi.
- Trust boundaries & attack surface: all data crossing Trust boundaries must be validated + authN/authZ; auditing to ensure accountability / Non-repudiation .
  mọi dữ liệu băng qua ranh giới tin cậy phải được validate + authN/authZ; auditing để đảm bảo trách nhiệm giải trình / không chối bỏ.
- Secure-SDLC Overlay across 5 activities; produce evidence in an Evidence Store.
  Lớp bảo mật phủ lên 5 hoạt động; tạo bằng chứng đưa vào kho Evidence..
- Quality Gates: Gates: SDR , Bug bar , FSR pass/fail by measurable criteria.
  Cổng chất lượng: SDR, Bug Bar, FSR — tiêu chí đo đếm rõ ràng.
- Traceability: REQ → design → code → tests → release → ops.
  Truy vết: Yêu cầu → thiết kế → mã → kiểm thử → phát hành → vận hành.