SOFTWARE DEVELOPMENT GOVERNANCE POLICY

CHÍNH SÁCH QUẢN TRỊ PHÁT TRIỂN PHẦN MỀM

Traceability • Security by Design • Quality • Control • Accountability

Tổ chức / Organization	[Tên tổ chức]
Chủ sở hữu / Document Owner	[Head of Engineering]
Phê duyệt / Approved by	Engineering Governance Council
Ngày hiệu lực / Effective Date	[DD/MM/YYYY]
Phiên bản / Version	1.0
Chu kỳ rà soát / Review Cycle	Hằng năm, hoặc khi có thay đổi đáng kể về công nghệ, pháp luật hoặc cơ cấu tổ chức
Tài liệu cấp trên / Parent Document	Engineering Constitution v1.0 (Tầng 1)
Phân loại / Classification	INTERNAL

SAMPLE DOCUMENT

A. Kiểm soát tài liệu

DOCUMENT CONTROL

Phiên bản	Ngày	Người soạn	Mô tả thay đổi
1.0	[DD/MM/YYYY]	[Tên / Vai trò]	Ban hành lần đầu

B. Mục lục nội dung

CONTENTS

- 1. Mục đích
- 2. Phạm vi áp dụng
- 3. Liên kết với Engineering Constitution
- 4. Định nghĩa
- 5. Yêu cầu bắt buộc
- 6. Vai trò và trách nhiệm
- 7. Đo lường tuân thủ
- 8. Cơ chế ngoại lệ
- 9. Không tuân thủ
- 10. Chủ sở hữu tài liệu và chu kỳ rà soát
- 11. Tài liệu liên quan
- 12. Phê duyệt

1. Mục đích

PURPOSE

1. Chuyển hóa Mười Nguyên tắc Nền tảng của Engineering Constitution thành các yêu cầu bắt buộc, cụ thể hơn, áp dụng cho toàn bộ hoạt động phát triển phần mềm.
2. Bảo đảm mọi sản phẩm/dịch vụ phần mềm được phát triển theo một vòng đời (SDLC) đã được phê duyệt, có thể kiểm tra và có bằng chứng tuân thủ.
3. Thiết lập ranh giới trách nhiệm rõ ràng giữa Product, Engineering, Quality, Security, Architecture và Operations trong suốt vòng đời phần mềm.
4. Làm nền tảng để ban hành các Standard và Procedure chi tiết hơn (Tầng 3, Tầng 4) mà không cần sửa đổi Policy mỗi khi công nghệ hoặc công cụ thay đổi.

Chính sách này không quy định chi tiết kỹ thuật (ngôn ngữ lập trình, framework, công cụ CI/CD cụ thể...) — các nội dung đó thuộc Tầng 3 (Standard) và Tầng 4 (Procedure).

2. Phạm vi áp dụng

SCOPE

- Toàn bộ phần mềm, dịch vụ, API và thành phần hệ thống do tổ chức tự phát triển, thuê ngoài, mua, cấu hình hoặc tích hợp.
- Toàn bộ vòng đời phần mềm: từ ý tưởng, phân tích yêu cầu, thiết kế, phát triển, kiểm thử, triển khai, vận hành, đến khi ngừng sử dụng (decommission).
- Áp dụng cho nhân sự nội bộ, nhà thầu, đối tác và nhà cung cấp tham gia hoạt động phát triển phần mềm.
- Áp dụng cho mọi mô hình phát triển (Agile/Scrum, Kanban, Waterfall, hybrid) — Policy này không bắt buộc một phương pháp luận cụ thể, nhưng bắt buộc phương pháp luận được chọn phải được ghi nhận chính thức trong tài liệu SDLC (xem Điều 5.1).

3. Liên kết với Engineering Constitution

ALIGNMENT WITH THE CONSTITUTION

Bảng dưới đây cho thấy mỗi nhóm yêu cầu bắt buộc trong Policy này trực tiếp hiện thực hóa nguyên tắc nào của Constitution — giúp người rà soát luôn truy vết được lý do đằng sau mỗi yêu cầu.

Yêu cầu trong Policy	Nguyên tắc Constitution tương ứng
5.1 Vòng đời phát triển phải được xác định	Điều 1 — Business Value First
5.2 Bảo mật, quyền riêng tư và phân tách môi trường	Điều 2 — Security and Privacy by Design
5.3 Chất lượng trong từng giai đoạn	Điều 3 — Quality Is Everyone's Responsibility
5.4 Kiểm soát thay đổi và phân tách trách nhiệm	Điều 4 — Segregation of Duties

5.5 Tự động hóa pipeline	Điều 5 — Everything as Code Where Practical
5.6 Khả năng quan sát và vận hành	Điều 6 — Observable and Operable by Default
5.7 Thay đổi nhỏ, có thể đảo ngược	Điều 7 — Small, Reversible Changes
5.8 Ghi nhận quyết định kiến trúc (ADR)	Điều 8 — Evidence over Opinion
5.9 Đánh giá tái sử dụng trước khi xây mới	Điều 9 — Reuse Before Build
5.10 Chuỗi cung ứng phần mềm và mã nguồn mở	Điều 2, khoản 6 — quản trị mã nguồn mở như chuỗi cung ứng
5.11 Phát triển thuê ngoài và bên thứ ba	Điều 2 — cùng một chuẩn bảo mật bất kể ai viết code
5.12 Đào tạo và nâng cao nhận thức	Điều 10 — Continuous Learning
5.13 Post-incident review và cải tiến	Điều 10 — Continuous Learning

4. Định nghĩa

DEFINITIONS

- **SDLC (Software Development Life Cycle):** tài liệu Tầng 4 (Procedure) mô tả các giai đoạn phát triển phần mềm, vai trò, gate chuyển giai đoạn, và bằng chứng/deliverable bắt buộc ở mỗi giai đoạn. Mỗi dự án/sản phẩm phải áp dụng một SDLC đã được phê duyệt theo Policy này.
- **Gate:** điểm kiểm soát bắt buộc giữa hai giai đoạn của SDLC, tại đó một vai trò có thẩm quyền xác nhận điều kiện chuyển giai đoạn đã được đáp ứng.
- **Định nghĩa hoàn tất (Definition of Done):** tập tiêu chí bắt buộc để một hạng mục công việc được coi là hoàn thành, tối thiểu gồm: đã kiểm thử, đã review, đáp ứng Coding/Secure Coding Standard, có tài liệu cần thiết.
- **Phân loại dữ liệu (Data Classification):** việc gán một hệ thống/tập dữ liệu vào một mức độ nhạy cảm (ví dụ: Công khai, Nội bộ, Bảo mật, Tối mật) theo Data Classification Standard — làm căn cứ xác định mức độ kiểm soát cần áp dụng.
- **Ngoại lệ (Exception):** cơ chế miễn trừ có kiểm soát, có thời hạn, đối với một yêu cầu cụ thể của Policy này — xem Điều 8.

5. Yêu cầu bắt buộc

MANDATORY REQUIREMENTS

Từ khóa MUST/SHOULD/MAY trong tài liệu này dùng theo đúng bảng quy ước tại Engineering Constitution (RFC 2119).

5.1 Vòng đời phát triển phần mềm (SDLC)

1. Mỗi dự án hoặc sản phẩm phần mềm **PHẢI (MUST)** áp dụng một SDLC bằng văn bản, đã được Engineering Governance Council phê duyệt hoặc phê duyệt theo ủy quyền.
2. SDLC được chọn **CÓ THỂ (MAY)** dựa trên bất kỳ phương pháp luận nào (Scrum, Kanban, Waterfall, hybrid...), nhưng **PHẢI (MUST)** xác định rõ: các giai đoạn, vai trò và trách nhiệm ở mỗi giai đoạn, gate bắt buộc, deliverable/bằng chứng cần lưu tại mỗi gate, và Definition of Done.
3. Dự án **KHÔNG ĐƯỢC PHÉP (MUST NOT)** bỏ qua giai đoạn kiểm thử hoặc giai đoạn rà soát bảo mật để rút ngắn thời gian, trừ khi có ngoại lệ được phê duyệt theo Điều 8.
4. SDLC **PHẢI (MUST)** được rà soát tối thiểu mỗi năm một lần bởi chủ sở hữu tài liệu.

5.2 Bảo mật, quyền riêng tư và phân tách môi trường

1. Yêu cầu bảo mật và quyền riêng tư **PHẢI (MUST)** được đưa vào từ giai đoạn phân tích/thiết kế của SDLC, không được bổ sung sau khi phát triển hoàn tất.
2. Mỗi hệ thống **PHẢI (MUST)** được phân loại mức độ nhạy cảm dữ liệu theo Data Classification Standard; mức phân loại này là căn cứ để xác định mức độ đánh giá rủi ro bảo mật/quyền riêng tư cần thực hiện trước khi triển khai production.
3. Môi trường Phát triển (Dev), Kiểm thử (Test) và Vận hành (Production) **PHẢI (MUST)** được phân tách độc lập về mặt logic hoặc vật lý. Dữ liệu production **KHÔNG ĐƯỢC PHÉP (MUST NOT)** được dùng ở môi trường Dev/Test trừ khi đã được ẩn danh hóa hoặc che dữ liệu (anonymized/masked) theo Data Protection Standard.
4. Chi tiết kỹ thuật (danh sách kiểm tra bảo mật, ngưỡng chấp nhận lỗ hổng...) do Secure Software Development Policy và Secure Coding Standard quy định.

5.3 Chất lượng trong từng giai đoạn

1. Mỗi giai đoạn của SDLC **PHẢI (MUST)** có tiêu chí chất lượng đầu ra, không chỉ kiểm tra ở cuối vòng đời.
2. Mọi thay đổi mã nguồn quan trọng **PHẢI (MUST)** được kiểm thử tự động hoặc thủ công tương xứng với rủi ro trước khi triển khai production.
3. Rủi ro chất lượng đã biết nhưng chưa xử lý **PHẢI (MUST)** được công khai và được người có thẩm quyền chấp nhận trước khi phát hành.

5.4 Kiểm soát thay đổi và phân tách trách nhiệm

1. Người tạo ra một thay đổi quan trọng **KHÔNG ĐƯỢC PHÉP (MUST NOT)** là người duy nhất phê duyệt thay đổi đó.
2. Mọi thay đổi trước khi được sáp nhập vào luồng mã chính (main codebase) **PHẢI (MUST)** được rà soát bởi ít nhất một người độc lập (peer review). Cơ chế kỹ thuật cụ thể để thực thi yêu cầu này do CI/CD Standard quy định.

3. Khi không thể phân tách đầy đủ (ví dụ đội ngũ quá nhỏ), **PHẢI (MUST)** có control thay thế (audit log, rà soát của quản lý, time-bound access).

5.5 Tự động hóa pipeline

1. Build, kiểm thử tự động và quét bảo mật cơ bản **NÊN (SHOULD)** được tích hợp vào pipeline CI/CD.
2. Hạ tầng và cấu hình quan trọng **NÊN (SHOULD)** được quản lý bằng version control (Infrastructure as Code).
3. Việc không tự động hóa một bước có thể tự động hóa được **PHẢI (MUST)** có lý do bằng văn bản và chủ sở hữu chịu trách nhiệm.

5.6 Khả năng quan sát và vận hành

1. Trước khi triển khai production, hệ thống **PHẢI (MUST)** có logging, monitoring và alerting ở mức tối thiểu để phát hiện sự cố và đo lường tình trạng hoạt động.
2. Hệ thống quan trọng (xếp hạng theo Standard liên quan) **PHẢI (MUST)** có runbook và kế hoạch rollback trước khi đưa vào production.
3. Log **KHÔNG ĐƯỢC PHÉP (MUST NOT)** chứa credentials, token hoặc dữ liệu cá nhân không cần thiết.

5.7 Quản lý thay đổi nhỏ, có thể đảo ngược

1. Triển khai production **NÊN (SHOULD)** hỗ trợ khả năng rollback hoặc roll-forward an toàn.
2. Thay đổi không thể đảo ngược (ví dụ migration dữ liệu một chiều) **PHẢI (MUST)** được đánh giá rủi ro riêng và có kế hoạch phục hồi trước khi thực hiện.

5.8 Quyết định dựa trên bằng chứng

1. Quyết định kiến trúc quan trọng (ảnh hưởng nhiều hệ thống, khó đảo ngược, hoặc tốn kém nếu sai) **PHẢI (MUST)** được ghi nhận bằng một định dạng quyết định kiến trúc chính thức (ví dụ Architecture Decision Record — ADR).
2. Bản ghi quyết định **PHẢI (MUST)** nêu rõ bối cảnh, các phương án đã xem xét, quyết định, lý do và hệ quả.

5.9 Tái sử dụng trước khi xây mới

1. Trước khi xây dựng một năng lực mới, đội dự án **PHẢI (MUST)** kiểm tra khả năng tái sử dụng thành phần dùng chung, dịch vụ hoặc API sẵn có trong tổ chức.
2. Quyết định "build" thay vì "reuse/buy/extend" **PHẢI (MUST)** được ghi nhận lý do theo cùng cơ chế tại Điều 5.8, vì đây cũng là một quyết định kiến trúc quan trọng.

5.10 Chuỗi cung ứng phần mềm và mã nguồn mở

1. Dự án **PHẢI (MUST)** quản lý rủi ro từ các thành phần mã nguồn mở và thư viện bên thứ ba, bao gồm việc duy trì danh mục thành phần phần mềm (Software Bill of Materials — SBOM) cho các hệ thống quan trọng.
2. Dependency **PHẢI (MUST)** được quét lỗ hổng định kỳ; lỗ hổng phát hiện **PHẢI (MUST)** được xử lý theo thời hạn (SLA) quy định tại Vulnerability Management Standard.

5.11 Phát triển thuê ngoài và bên thứ ba

1. Phần mềm do đối tác, nhà thầu hoặc đội ngũ thuê ngoài phát triển **PHẢI (MUST)** tuân thủ các yêu cầu bảo mật tương đương với phát triển nội bộ theo Policy này.
2. Bàn giao từ bên thứ ba **PHẢI (MUST)** được đánh giá bảo mật độc lập trước khi đưa vào production, tương xứng với mức độ nhạy cảm của hệ thống.

5.12 Đào tạo và nâng cao nhận thức

1. Nhân sự tham gia SDLC **PHẢI (MUST)** hoàn thành đào tạo nhận thức bảo mật (bao gồm secure coding phù hợp vai trò) theo chu kỳ do tổ chức quy định.
2. Hồ sơ đào tạo **NÊN (SHOULD)** được lưu giữ làm bằng chứng tuân thủ khi có yêu cầu kiểm toán.

5.13 Học hỏi liên tục

1. Sự cố production nghiêm trọng **PHẢI (MUST)** có post-incident review tập trung vào nguyên nhân hệ thống, theo tinh thần blameless (không quy trách nhiệm cá nhân), trừ trường hợp có hành vi cố ý vi phạm.
2. Mỗi post-incident review **PHẢI (MUST)** tạo ra ít nhất một hành động cải tiến có chủ sở hữu và thời hạn cụ thể.

5.14 Phát triển phần mềm có hỗ trợ của AI

1. Việc sử dụng công cụ AI tạo sinh (AI code assistant, AI agent...) trong phát triển phần mềm **PHẢI (MUST)** tuân theo AI-Assisted Software Development Policy hiện hành.
2. Mã nguồn hoặc cấu hình do AI tạo ra **PHẢI (MUST)** được con người review trước khi sáp nhập vào luồng mã chính — không được coi là "đã được review" chỉ vì đã qua một công cụ AI khác kiểm tra.

> **Lưu ý về mức độ NÊN (SHOULD):** Việc không tuân theo một yêu cầu ở mức NÊN (SHOULD) không bắt buộc phải qua quy trình ngoại lệ hình thức tại Điều 8, nhưng phải được ghi nhận lý do và được người quản lý trực tiếp chấp thuận. Quy trình ngoại lệ hình thức tại Điều 8 áp dụng bắt buộc cho các yêu cầu ở mức PHẢI (MUST).

6. Vai trò và trách nhiệm

ROLES & RESPONSIBILITIES

- **Engineering Governance Council:** phê duyệt hoặc ủy quyền phê duyệt SDLC; phê duyệt sửa đổi Policy này; phê duyệt ngoại lệ rủi ro cao.
- **Chủ sở hữu tài liệu (Document Owner — [Head of Engineering]):** duy trì nội dung Policy, đảm bảo rà soát đúng chu kỳ, tiếp nhận phản hồi từ các đội.
- **Engineering Leadership:** chuyển yêu cầu của Policy thành Standard/Procedure cụ thể; cung cấp công cụ, nền tảng và chương trình đào tạo giúp các đội tuân thủ dễ dàng.
- **Product/Project Owner:** đảm bảo dự án áp dụng đúng SDLC đã phê duyệt ngay từ giai đoạn khởi tạo.
- **Engineering Teams:** thực thi Policy trong công việc hằng ngày; hoàn thành đào tạo bắt buộc theo Điều 5.12; báo cáo khi phát hiện yêu cầu không khả thi.
- **Architecture, Security, Risk & Compliance:** thẩm định các gate liên quan đến kiến trúc, bảo mật, rủi ro trong SDLC; tham gia post-incident review.

7. Đo lường tuân thủ

COMPLIANCE MEASUREMENT

Mức độ tuân thủ Policy này được đo lường định kỳ (ngưỡng đo cụ thể do Standard liên quan quy định), tối thiểu gồm:

- Tỷ lệ dự án có SDLC được phê duyệt chính thức.
- Tỷ lệ thay đổi mã nguồn có ít nhất một reviewer độc lập.
- Tỷ lệ hệ thống quan trọng có SBOM và kết quả quét dependency còn hiệu lực.
- Tỷ lệ sự cố nghiêm trọng có post-incident review và hành động cải tiến đã đóng.
- Tỷ lệ nhân sự hoàn thành đào tạo bảo mật đúng hạn.
- Số lượng ngoại lệ đang mở hoặc quá hạn.

8. Cơ chế ngoại lệ

EXCEPTIONS

Ngoại lệ đối với bất kỳ yêu cầu PHẢI (MUST) nào trong Policy này phải theo đúng quy trình và giới hạn của Engineering Constitution, Điều 8, bao gồm tối thiểu: quy định cần miễn trừ, lý do, phạm vi và thời hạn, rủi ro và control thay thế, người chịu trách nhiệm, người phê duyệt, ngày hết hiệu lực và kế hoạch quay lại tuân thủ.

Không được cấp ngoại lệ trái với các giới hạn tuyệt đối đã nêu tại Constitution (yêu cầu pháp luật, hành vi gian lận, tự phê duyệt thay đổi quan trọng mà không có control thay thế...).

9. Không tuân thủ

NON-COMPLIANCE

Vi phạm Policy này mà không có ngoại lệ được phê duyệt **PHẢI (MUST)** được báo cáo cho Engineering Leadership hoặc Engineering Governance Council. Tùy mức độ, hậu quả có thể bao gồm yêu cầu khắc phục

trong thời hạn xác định, hoặc tạm dừng triển khai production đối với hệ thống liên quan.

Vi phạm mang tính **cố ý** — như cố tình vượt qua gate kiểm soát đã được phê duyệt, che giấu sự cố hoặc làm giả bằng chứng tuân thủ — **CÓ THỂ (MAY)** được xem xét xử lý theo Nội quy lao động và các điều khoản hợp đồng hiện hành, dưới sự phối hợp của bộ phận Nhân sự (HR) và Pháp chế (Legal). Policy này không tự thân là căn cứ kỷ luật lao động; mọi hình thức kỷ luật phải tuân theo quy định pháp luật lao động và văn bản nội bộ có hiệu lực tương ứng.

Quy định này không mâu thuẫn với văn hóa blameless tại Điều 5.13 — post-incident review vẫn tập trung vào nguyên nhân hệ thống; chỉ hành vi cố ý nêu trên mới thuộc phạm vi xem xét kỷ luật.

10. Chủ sở hữu tài liệu và chu kỳ rà soát

DOCUMENT OWNERSHIP & REVIEW CYCLE

Document Owner: [Head of Engineering]. Chu kỳ rà soát: hằng năm, hoặc ngay khi có thay đổi đáng kể về công nghệ, quy định pháp luật hoặc cơ cấu tổ chức. Mọi thay đổi Policy này phải được Engineering Governance Council phê duyệt trước khi ban hành phiên bản mới.

11. Tài liệu liên quan

RELATED DOCUMENTS

- Engineering Constitution v1.0 (Tầng 1)
- Secure Software Development Policy, AI-Assisted Software Development Policy, Data Protection in Software Development Policy (Tầng 2)
- Secure Coding Standard, Testing Standard, CI/CD Standard, Data Classification Standard, Vulnerability Management Standard (Tầng 3)
- [Tên SDLC cụ thể của tổ chức], Production Release Procedure, Exception Management Procedure (Tầng 4)

Mỗi dự án có một vòng đời được xác định rõ;
bảo mật và chất lượng được xây từ thiết kế, không vá sau;
mọi thay đổi quan trọng có người review độc lập;
mọi quyết định kiến trúc có bằng chứng và lý do;
mọi ngoại lệ đều có thời hạn và người chịu trách nhiệm;
và mọi sự cố đều trở thành một bài học được hành động.

12. Phê duyệt

APPROVAL

Vai trò	Họ tên	Chữ ký	Ngày
Chair, Engineering Governance Council			
Chief Technology Officer			
Head of Engineering (Document Owner)			
Head of Technology Risk / Security			