CI/CD STANDARD

TIÊU CHUẨN KỸ THUẬT — TIÊU CHUẨN CI/CD

Build Once, Promote Everywhere • Pipeline Is Code Too • Fail Fast, Fail Loud

Tổ chức / Organization	[Tên tổ chức]
Chủ sở hữu / Document Owner	Head of DevOps / Platform 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 ngay khi công cụ CI/CD chính của tổ chức thay đổi đáng kể hoặc sau sự cố nghiêm trọng liên quan đến pipeline
Tài liệu cấp trên / Parent Documents	Engineering Constitution v1.0 (Tầng 1); Software Development Governance Policy v1.0 (Tầng 2)
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 Policy liên quan
- 4. Định nghĩa
- 5. Yêu cầu kỹ thuật bắt buộc
- 6. Công cụ khuyến nghị
- 7. Đo lường tuân thủ
- 8. Cơ chế ngoại lệ
- 9. Chủ sở hữu tài liệu và chu kỳ rà soát
- 10. Tài liệu liên quan
- 11. Phê duyệt

1. Mục đích

PURPOSE

1. Cụ thể hóa Software Development Governance Policy (SDGP), Điều 5.5 — Tự động hóa pipeline — thành các quy tắc kỹ thuật CỤ THỂ, ĐO LƯỜNG ĐƯỢC về cấu trúc và hành vi của pipeline CI/CD, thay vì chỉ nêu nguyên tắc "phải tự động hóa" ở mức Policy.
2. Bảo đảm mọi thay đổi mã nguồn đi vào production đều qua một pipeline nhất quán, có thể tái lập (reproducible) và có thể kiểm tra (auditable), làm nền tảng kỹ thuật để Change and Release Management Policy (CRMP) thực thi được các yêu cầu về versioning và rollback.
3. Đặt nguyên tắc "build once, deploy many" (artifact chỉ được xây dựng đúng một lần rồi thăng cấp qua các môi trường) làm nguyên lý trung tâm của Standard này.

Standard này giả định người đọc đã đồng ý với các nguyên tắc tại SDGP và CRMP — không lặp lại lý lẽ "vì sao", chỉ quy định CHÍNH XÁC pipeline phải có cấu trúc và hành vi kỹ thuật gì.

2. Phạm vi áp dụng

SCOPE

- Áp dụng cho pipeline CI/CD của mọi phần mềm/dịch vụ do tổ chức phát triển và vận hành, đưa mã nguồn hoặc artifact vào bất kỳ môi trường nào từ dev trở lên — không chỉ riêng production.
- Áp dụng cho mọi công cụ CI/CD tổ chức đang sử dụng; Standard này trung lập về công cụ và trung lập về mô hình vận hành (pipeline có thể theo mô hình đẩy — `push-based`, kích hoạt job đưa thay đổi tới môi trường đích — hoặc mô hình kéo kiểu GitOps — `pull-based`, nơi một agent trong môi trường đích tự đồng bộ trạng thái theo cấu hình khai báo trong version control; cả hai mô hình đều PHẢI tuân thủ các yêu cầu tại Điều 5).
- **KHÔNG quy định lại** nội dung của từng loại kiểm tra bảo mật (SAST/DAST/SCA — xem Secure Software Development Policy), cách sinh SBOM (xem Open-Source Software Policy), hay điều kiện phân loại rủi ro thay đổi/phê duyệt/rollback chi tiết (xem CRMP). Standard này chỉ quy định pipeline phải có ĐÚNG CHỖ ĐỨNG (gate) cho các kiểm tra đó và phải xử lý đúng cách khi chúng thất bại.

3. Liên kết với Policy liên quan

ALIGNMENT WITH RELATED POLICIES

Yêu cầu trong Standard	Điều khoản Policy/Constitution tương ứng
5.1 Pipeline-as-code, versioned	SDGP Điều 5.5; Engineering Constitution Điều 5 — Everything as Code
5.2 Pull request + review cho thay đổi pipeline	SDGP Điều 5.4 — Segregation of Duties
5.3 Build once, promote many	SDGP Điều 5.5; CRMP Điều 5.3 — Versioning và bất biến artifact
5.4 Artifact versioned và immutable	CRMP Điều 4, Điều 5.3 — Semantic Versioning, bất biến

5.5 Giai đoạn tuần tự bắt buộc và quy tắc chặn	SDGP Điều 5.5; Secure Software Development Policy Điều 5.3; Open-Source Software Policy Điều 5.2-5.3
5.6 Quản lý secret trong pipeline	Secure Software Development Policy Điều 5.5
5.7 Log và khả năng tự chẩn đoán	Engineering Constitution Điều 6 — Observable and Operable by Default
5.8 Hỗ trợ rollback/roll-forward khi triển khai	CRMP Điều 5.4 — Rollback và roll-forward

4. Định nghĩa

DEFINITIONS

- **Pipeline CI/CD:** chuỗi các giai đoạn tự động (build, test, kiểm tra chất lượng/bảo mật, đóng gói, triển khai) được kích hoạt khi có thay đổi mã nguồn hoặc theo lịch, đưa thay đổi từ version control đến môi trường chạy thực tế.
- **Pipeline-as-Code:** thực hành định nghĩa toàn bộ cấu hình pipeline (các giai đoạn, thứ tự, điều kiện gate) bằng file cấu hình dạng văn bản, có cấu trúc (ví dụ YAML), lưu trong version control cùng mã nguồn hoặc repository cấu hình quản lý tương đương, thay vì cấu hình thủ công qua giao diện web của công cụ CI/CD. Đây là một thực hành phổ biến rộng rãi trong ngành (không thuộc một tổ chức chuẩn hóa chính thức nào), thuộc phong trào "Everything as Code" đã nêu tại Engineering Constitution Điều 5.
- **Artifact:** sản phẩm build hoàn chỉnh, có thể triển khai được (ví dụ: container image, package, gói cài đặt đóng gói), sinh ra từ một lần build mã nguồn tại một commit cụ thể.
- **Build Once, Deploy Many / Artifact Immutability:** nguyên tắc — có nguồn gốc từ nguyên lý "Build your binaries only once" trong cuốn sách kinh điển *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation* (Jez Humble & David Farley) — theo đó một artifact được build đúng một lần rồi cùng một bản build đó được thăng cấp (promote) tuần tự qua các môi trường, KHÔNG build lại từ mã nguồn cho từng môi trường; artifact đã publish không bị sửa đổi lại (immutable).
- **Promote (thăng cấp môi trường):** hành động chuyển cùng một artifact đã build từ môi trường này sang môi trường kế tiếp (ví dụ staging → production) sau khi đạt các gate yêu cầu, không build lại.
- **Environment Parity (sự đồng nhất môi trường):** mức độ giống nhau về hạ tầng, cấu hình nền tảng và phiên bản phần mềm hệ thống giữa các môi trường (dev/staging/production), khác biệt chỉ nên nằm ở quy mô (scale) và giá trị cụ thể của biến môi trường/secret — điều kiện cần để "build once, deploy many" thực sự an toàn (artifact chạy đúng ở production như đã chạy ở staging).
- **Quality Gate / Security Gate:** điểm kiểm soát tự động trong pipeline có khả năng chặn (block) giai đoạn kế tiếp khi điều kiện không đạt — chi tiết ngưỡng/loại kiểm tra do Standard/Policy chuyên biệt quy định (Secure Software Development Policy, Open-Source Software Policy).
- **Job thông tin (Informational/Warn-only Job):** một bước kiểm tra trong pipeline được cấu hình chỉ để cảnh báo/thu thập dữ liệu, không có quyền chặn (block) giai đoạn kế tiếp — dùng khi thử nghiệm một công cụ/kiểm tra mới trước khi chính thức biến nó thành một Quality/Security Gate bắt buộc.
- **Secret Manager:** hệ thống chuyên dụng lưu trữ, cấp phát secret có kiểm soát cho pipeline — định nghĩa đầy đủ về secret/credential tại Secure Software Development Policy, Điều 4.

- **DORA metrics:** bốn chỉ số deployment frequency, lead time for change, change failure rate, và mean time to restore (MTTR) — định nghĩa đầy đủ tại CRMP, Điều 4. Standard này chỉ dẫn chiếu, không định nghĩa lại, và giữ nguyên đúng bốn chỉ số này để nhất quán trong toàn bộ series tài liệu, kể cả khi nghiên cứu DORA gần đây có bổ sung thêm chỉ số khác.
- **Rollback / Roll-forward:** định nghĩa đầy đủ tại CRMP, Điều 4.

5. Yêu cầu kỹ thuật bắt buộc

MANDATORY TECHNICAL REQUIREMENTS

5.1 Pipeline-as-code bắt buộc

1. Toàn bộ pipeline CI/CD PHẢI (MUST) được định nghĩa dưới dạng file cấu hình versioned (ví dụ YAML) lưu trong cùng repository mã nguồn hoặc một repository cấu hình được quản lý tương đương.
2. KHÔNG ĐƯỢC PHÉP (MUST NOT) có bước build/deploy quan trọng chỉ tồn tại dưới dạng cấu hình thủ công qua giao diện web (UI) của công cụ CI/CD mà không có bản sao trong code — bất kỳ thiết lập nào ảnh hưởng đến kết quả pipeline (biến môi trường, thứ tự stage, điều kiện gate) PHẢI truy vết được trong version control.
3. Trường hợp công cụ CI/CD bắt buộc một số cấu hình toàn cục chỉ cấu hình được qua UI (ví dụ quyền truy cập hạ tầng, kết nối tới cụm triển khai), phần đó PHẢI được ghi lại thành tài liệu và rà soát định kỳ, xem như một ngoại lệ có kiểm soát chứ không phải hành vi mặc định của tổ chức.

5.2 Thay đổi pipeline phải qua pull request

1. Mọi thay đổi vào file định nghĩa pipeline PHẢI (MUST) đi qua pull request và được review, theo đúng quy trình review mã nguồn thông thường và nguyên tắc phân tách trách nhiệm tại SDGP Điều 5.4 — người review PHẢI độc lập với người đề xuất thay đổi.
2. KHÔNG ĐƯỢC PHÉP (MUST NOT) tự ý sửa trực tiếp pipeline đang chạy cho môi trường production (hoặc nhánh chính) mà không qua pull request/review, kể cả khi người sửa có quyền quản trị công cụ CI/CD.
3. Thay đổi pipeline ảnh hưởng đến một Quality/Security Gate bắt buộc (Điều 5.5) NÊN (SHOULD) có thêm review của một người có thẩm quyền về bảo mật/chất lượng liên quan, tương tự yêu cầu rà soát bổ sung cho thay đổi ảnh hưởng cao tại CRMP Điều 5.2.4.

5.3 Build once, promote many

1. Artifact triển khai PHẢI (MUST) được build đúng một lần từ một điểm mã nguồn (commit) cụ thể; cùng một artifact đó PHẢI được thăng cấp (promote) tuần tự qua các môi trường (ví dụ dev → staging → production), theo đúng nguyên tắc "build once, deploy many" tại Điều 4.
2. KHÔNG ĐƯỢC PHÉP (MUST NOT) build lại từ mã nguồn riêng biệt cho từng môi trường (ví dụ build lại khi deploy lên production sau khi đã build và test ở staging) — mục tiêu là bảo đảm artifact đã kiểm thử ở staging chính là artifact chạy ở production, không phải một bản build "tương tự cùng mã nguồn".
3. Cấu hình đặc thù theo môi trường (biến môi trường, endpoint, feature flag) PHẢI được tách khỏi artifact và nạp vào tại thời điểm triển khai/khởi động, không đóng cứng (hardcode) vào artifact theo từng môi trường — đây là điều kiện kỹ thuật để build-once khả thi.
4. Để nguyên tắc build-once thực sự bảo đảm "staging chạy đúng như production", các môi trường PHẢI (MUST) đạt environment parity ở mức tối thiểu về hệ điều hành/runtime nền tảng và phiên bản các thành phần hạ tầng phụ thuộc (ví dụ phiên bản database, message broker); khác biệt được chấp nhận chỉ nên nằm ở quy mô (scale) và giá trị biến môi trường/secret.

5.4 Artifact phải được versioned và immutable

1. Mỗi artifact PHẢI (MUST) được gắn một định danh phiên bản duy nhất, truy vết được về đúng commit nguồn đã build ra nó, tuân thủ định dạng phiên bản đã quy định tại CRMP Điều 4 (Semantic Versioning)

đối với artifact có API công khai.

2. Artifact đã publish vào kho lưu trữ artifact (artifact registry/repository) PHẢI (MUST) là bất biến — KHÔNG ĐƯỢC PHÉP (MUST NOT) ghi đè nội dung lên một phiên bản đã publish; cần sửa thì phải build và publish một phiên bản mới.
3. Kho lưu trữ artifact NÊN (SHOULD) có cơ chế kỹ thuật (ví dụ khóa tag, chặn overwrite) để ngăn ghi đè ngoài ý muốn, không chỉ dựa vào quy ước làm việc giữa các kỹ sư.

5.5 Các giai đoạn tuần tự bắt buộc và quy tắc chặn

#	Giai đoạn	Vai trò
1	Build	Biên dịch/đóng gói mã nguồn thành artifact tạm thời để kiểm thử.
2	Kiểm thử tự động	Unit test, integration test theo yêu cầu SDGP.
3	Quality/Security Gate	SAST/SCA/DAST (SSDP Điều 5.3), quét thành phần mã nguồn mở/SBOM (OSSP Điều 5.2-5.3).
4	Đóng gói artifact	Sinh artifact chính thức, versioned, publish vào registry (Điều 5.4).
5	Triển khai	Promote artifact đã đóng gói vào môi trường đích (Điều 5.3, 5.8).

1. Pipeline PHẢI (MUST) bao gồm tối thiểu năm giai đoạn tuần tự trên, theo đúng thứ tự phụ thuộc logic (không đóng gói artifact chính thức khi chưa qua test/gate).
2. Mỗi giai đoạn hoặc Quality/Security Gate được xác lập là **bắt buộc (Mandatory Gate)** nếu thất bại PHẢI (MUST) chặn giai đoạn kế tiếp — KHÔNG ĐƯỢC PHÉP (MUST NOT) có cơ chế ngầm bỏ qua (silent skip) một Mandatory Gate thất bại để pipeline tiếp tục chạy, trừ khi đi qua cơ chế ngoại lệ tường minh, có phê duyệt và được ghi log (xem Điều 8).
3. Khi thử nghiệm một công cụ/kiểm tra mới, tổ chức ĐƯỢC PHÉP (MAY) cấu hình nó tạm thời như một **Job thông tin (informational/warn-only)** theo định nghĩa tại Điều 4 — không có quyền chặn pipeline — để thu thập dữ liệu trước khi chính thức nâng cấp thành Mandatory Gate; Job thông tin KHÔNG ĐƯỢC PHÉP (MUST NOT) thay thế một Mandatory Gate đã quy định tại SSDP/OSSP.
4. Trong cùng một giai đoạn logic, tổ chức ĐƯỢC PHÉP (MAY) chạy song song nhiều kiểm tra (ví dụ nhiều loại test), miễn giai đoạn đó chỉ được coi là hoàn tất khi toàn bộ nhánh song song thuộc diện Mandatory Gate đã đạt yêu cầu.

5.6 Quản lý secret trong pipeline

1. Secret dùng trong pipeline (credentials, API key, token triển khai...) PHẢI (MUST) được lưu trong secret manager của công cụ CI/CD hoặc dịch vụ quản lý secret riêng, theo đúng yêu cầu tại Secure Software Development Policy, Điều 5.5 — Standard này không quy định lại chi tiết, chỉ khẳng định pipeline là một trong các nơi bắt buộc áp dụng.
2. KHÔNG ĐƯỢC PHÉP (MUST NOT) lưu secret ở dạng plaintext trong file cấu hình pipeline (kể cả khi file đó nằm trong version control) hay trong biến môi trường cấu hình tĩnh, không mã hóa, đi kèm định nghĩa pipeline.
3. Log của pipeline PHẢI (MUST) tự động che (mask) giá trị secret đã khai báo qua secret manager, không hiển thị nguyên văn ra output/log ngay cả khi bị in ra do lỗi lập trình trong script pipeline.

5.7 Log và khả năng tự chẩn đoán

1. Pipeline PHẢI (MUST) cung cấp log/kết quả rõ ràng cho mỗi giai đoạn thất bại, tối thiểu bao gồm: giai đoạn nào thất bại, thông điệp lỗi cụ thể, và (nếu áp dụng) đường link đến chi tiết kết quả kiểm tra liên quan.
2. Log/kết quả đó PHẢI (MUST) đủ để kỹ sư của đội phát triển tự chẩn đoán nguyên nhân mà không cần quyền truy cập sâu vào hạ tầng vận hành công cụ CI/CD (ví dụ không cần quyền quản trị hệ thống CI/CD để xem log build của chính đội mình).
3. Pipeline NÊN (SHOULD) giữ lại log/kết quả trong một thời gian tối thiểu đủ để phục vụ điều tra sự cố và rà soát tuân thủ — thời hạn cụ thể do tổ chức tự quy định tương xứng với yêu cầu lưu trữ bằng chứng của tổ chức.

5.8 Hỗ trợ rollback/roll-forward và triển khai lũy tiến

1. Giai đoạn triển khai (deploy) PHẢI (MUST) được thiết kế kỹ thuật để hỗ trợ rollback (quay lại artifact/phiên bản trước) hoặc roll-forward (triển khai tiếp một bản sửa lỗi) nhanh, đúng theo yêu cầu về phương án rollback/roll-forward bắt buộc tại CRMP Điều 5.4 — Standard này không lặp lại điều kiện phê duyệt của CRMP, chỉ bảo đảm pipeline có khả năng kỹ thuật để thực hiện.
2. KHÔNG ĐƯỢC PHÉP (MUST NOT) thiết kế pipeline chỉ hỗ trợ triển khai một chiều tiến (forward-only) khiến việc quay lại phiên bản ổn định trước đó đòi hỏi thao tác thủ công phức tạp hoặc build lại từ đầu — nhờ artifact đã versioned và immutable (Điều 5.4), rollback về bản chất là triển khai lại một artifact cũ đã có sẵn trong registry.
3. Đối với môi trường production, pipeline NÊN (SHOULD) hỗ trợ chiến lược triển khai lũy tiến (progressive delivery — ví dụ canary release, blue/green deployment, hoặc dùng feature flag để bật dần tính năng) nhằm giới hạn phạm vi ảnh hưởng (blast radius) của một thay đổi lỗi trước khi cần đến rollback toàn phần.
4. Với thay đổi có migration dữ liệu một chiều không thể rollback kỹ thuật thuần túy, pipeline NÊN (SHOULD) hỗ trợ triển khai theo giai đoạn có thể dừng giữa chừng, theo đúng khuyến nghị kỹ thuật "expand/contract" đã nêu tại CRMP Điều 5.4.2.

> **Không lặp lại nội dung của SSDP/OSSP/CRMP:** Chi tiết công cụ và ngưỡng của SAST/DAST/SCA thuộc Secure Software Development Policy; chi tiết định dạng và nội dung SBOM thuộc Open-Source Software Policy; điều kiện phê duyệt thay đổi, phân loại rủi ro và versioning chi tiết thuộc Change and Release Management Policy. Standard này chỉ quy định pipeline phải có đúng cấu trúc để các quy định đó được thực thi đúng chỗ.

6. Công cụ khuyến nghị

RECOMMENDED TOOLING

- Công cụ CI/CD hỗ trợ pipeline-as-code dạng cấu hình versioned (ví dụ GitHub Actions, GitLab CI, Jenkins declarative pipeline/ `Jenkinsfile`, Azure DevOps YAML pipelines, hoặc công cụ GitOps như ArgoCD/Flux cho phần CD theo mô hình pull-based) — tổ chức tự lựa chọn theo hạ tầng hiện có, Standard này không ràng buộc một công cụ cụ thể.
- Artifact registry hỗ trợ immutable tag/versioning (ví dụ container registry có tính năng khóa tag hoặc chặn ghi đè).
- Secret manager tích hợp CI/CD theo đúng danh mục đã phê duyệt tại Secure Software Development Policy.
- Công cụ observability/log tập trung cho kết quả pipeline, bảo đảm kỹ sư tự tra cứu được mà không cần quyền quản trị hạ tầng CI/CD (Điều 5.7).

7. Đo lường tuân thủ

COMPLIANCE MEASUREMENT

- Tỷ lệ dự án có toàn bộ pipeline định nghĩa dưới dạng code trong version control (không có cấu hình quan trọng chỉ tồn tại trên UI).
- Tỷ lệ thay đổi vào file định nghĩa pipeline đi qua pull request/review đầy đủ.
- Tỷ lệ bản triển khai production dùng đúng artifact đã build và test ở staging (không build lại riêng cho production).
- Số lần phát hiện artifact bị ghi đè hoặc thiếu versioning trong artifact registry.
- Tỷ lệ Mandatory Gate thất bại nhưng vẫn để giai đoạn kế tiếp chạy tiếp (silent skip) — mục tiêu bằng 0 ngoài các trường hợp đã qua cơ chế ngoại lệ hợp lệ.
- Số lần phát hiện secret dạng plaintext trong cấu hình pipeline qua secret scanning (liên kết Secure Software Development Policy, Điều 5.5.3).
- Thời gian trung bình để rollback thành công một bản triển khai — đối chiếu với chỉ số MTTR trong bốn chỉ số DORA tại CRMP Điều 5.7.

8. Cơ chế ngoại lệ

EXCEPTIONS

Áp dụng theo đúng cơ chế và giới hạn tại Engineering Constitution, Điều 8 và SDGP, Điều 8. Vai trò/ trách nhiệm tổng quát và xử lý không tuân thủ áp dụng theo đúng SDGP, Điều 6 và Điều 9 — Standard này không lặp lại, chỉ nêu ngoại lệ đặc thù:

- Việc tạm thời bỏ qua một Mandatory Gate bắt buộc tại Điều 5.5 PHẢI (MUST) được Head of DevOps/ Platform Engineering hoặc người được ủy quyền phê duyệt bằng văn bản, có thời hạn cụ thể, và được ghi log tự động trong chính pipeline.
- Việc triển khai một pipeline chỉ hỗ trợ forward-only tạm thời (chưa đáp ứng Điều 5.8) chỉ được chấp nhận như một ngoại lệ có thời hạn, kèm kế hoạch khắc phục cụ thể, được phê duyệt theo cùng cơ chế trên.

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

DOCUMENT OWNERSHIP & REVIEW CYCLE

Document Owner: Head of DevOps / Platform Engineering. Chu kỳ rà soát: hằng năm, hoặc ngay khi công cụ CI/CD chính của tổ chức thay đổi đáng kể (ví dụ chuyển đổi nền tảng CI/CD, áp dụng mô hình GitOps quy mô lớn hơn) hoặc sau sự cố nghiêm trọng liên quan đến pipeline.

10. Tài liệu liên quan

RELATED DOCUMENTS

- Engineering Constitution v1.0, Software Development Governance Policy v1.0 (Tầng 1-2)
- Change and Release Management Policy, Secure Software Development Policy, Open-Source Software Policy (Tầng 2)
- Secure Coding Standard, Vulnerability Management Standard, Incident Management Standard (Tầng 3)
- Pipeline Configuration Guideline, Secret Management Procedure, Emergency Change Procedure (Tầng 4/5)

Một pipeline không đáng tin chỉ vì nó tự động — nó đáng tin vì cấu hình của nó nằm trong version control, mọi thay đổi đều qua review, và artifact chạy ở production là chính artifact đã được kiểm chứng ở staging, không phải một bản build khác mang cùng cái tên.

11. Phê duyệt

APPROVAL

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