ENGINEERING CONSTITUTION

HIẾN CHƯƠNG KỸ THUẬT

Business value • Trust • Quality • Control • Learning

Tổ chức / Organization	[Tên tổ chức]
Chủ sở hữu / Document Owner	[Head of Engineering]
Phê duyệt / Approved by	Technology Governance Council
Ngày hiệu lực / Effective Date	[DD/MM/YYYY]
Phiên bản / Version	1.0
Chu kỳ rà soát / Review Cycle	Ba năm một lần hoặc khi có thay đổi chiến lược trọng yếu
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. Lời mở đầu
- 2. Mục đích
- 3. Phạm vi áp dụng
- 4. Vị trí trong hệ thống quản trị
- 5. Mười nguyên tắc nền tảng
- 6. Nguyên tắc giải quyết xung đột
- 7. Quyền tự chủ và trách nhiệm
- 8. Ngoại lệ
- 9. Quản trị và trách nhiệm thực thi
- 10. Đánh giá việc áp dụng
- 11. Sửa đổi Hiến chương
- 12. Cam kết kỹ thuật
- 13. Phê duyệt

1. Lời mở đầu

PREAMBLE

Chúng ta xây dựng và vận hành các sản phẩm, nền tảng và dịch vụ công nghệ nhằm tạo ra giá trị bền vững cho khách hàng, Business Units và tổ chức.

Chúng ta tin rằng kỹ thuật phần mềm tốt không chỉ được thể hiện qua mã nguồn. Nó được thể hiện trong cách chúng ta hiểu vấn đề kinh doanh, đưa ra quyết định, thiết kế giải pháp, bảo vệ dữ liệu, kiểm soát thay đổi, vận hành hệ thống và học hỏi từ thực tế.

Hiến chương Kỹ thuật này xác lập các nguyên tắc tối cao áp dụng cho toàn bộ hoạt động phát triển và vận hành phần mềm của tổ chức.

Mọi Engineering Policy, Standard, Procedure, Guideline, Playbook và quyết định kỹ thuật phải phù hợp với Hiến chương này. Khi có mâu thuẫn, Hiến chương được ưu tiên áp dụng, trừ trường hợp pháp luật, yêu cầu của cơ quan quản lý hoặc chính sách cấp tập đoàn quy định ở mức cao hơn.

2. Mục đích

PURPOSE

1. Thiết lập một hệ giá trị và nguyên tắc chung cho toàn bộ cộng đồng kỹ thuật.
2. Định hướng các quyết định trong suốt vòng đời sản phẩm và phần mềm.
3. Cân bằng giữa giá trị kinh doanh, tốc độ, chất lượng, bảo mật, khả năng vận hành và chi phí.
4. Xác lập nền tảng cho hệ thống Engineering Policies, Standards và Procedures.
5. Trao quyền cho các nhóm kỹ thuật trong một khuôn khổ trách nhiệm rõ ràng.
6. Bảo đảm các quyết định kỹ thuật có thể được giải thích, kiểm chứng và truy vết.
7. Xây dựng văn hóa kỹ thuật dựa trên sự chính trực, hợp tác, bằng chứng và học hỏi liên tục.

Hiến chương không quy định chi tiết phải sử dụng công nghệ, công cụ hoặc phương pháp phát triển cụ thể. Những nội dung này được quy định tại các tài liệu quản trị cấp dưới.

3. Phạm vi áp dụng

SCOPE

- Tất cả sản phẩm, hệ thống, ứng dụng, API, nền tảng dữ liệu và dịch vụ công nghệ.
- Phần mềm do tổ chức tự phát triển, thuê ngoài, mua, cấu hình hoặc tích hợp.
- Toàn bộ vòng đời sản phẩm và phần mềm, từ ý tưởng đến ngừng sử dụng.
- Tất cả môi trường phát triển, kiểm thử, UAT, staging, production và disaster recovery.
- Toàn bộ nhân sự nội bộ, nhà thầu, đối tác và nhà cung cấp tham gia hoạt động kỹ thuật.
- Tất cả mô hình triển khai, gồm on-premises, cloud, hybrid cloud và managed services.

Các vai trò thuộc phạm vi bao gồm Product, Business Analysis, Architecture, Engineering, Quality, DevOps, Platform, SRE, Cybersecurity, Risk, Compliance, Data, AI, Delivery và các nhà cung cấp liên quan.

4. Vị trí trong hệ thống quản trị

GOVERNANCE HIERARCHY

Tầng	Loại tài liệu	Vai trò
Cấp 1	**Engineering Constitution**	Triết lý, giá trị và các nguyên tắc tối cao; ổn định trong nhiều năm.
Cấp 2	**Engineering Policies**	Các yêu cầu bắt buộc theo từng lĩnh vực quản trị.

Tầng	Loại tài liệu	Vai trò
Cấp 3	**Engineering Standards**	Yêu cầu kỹ thuật cụ thể, có thể kiểm tra hoặc đo lường.
Cấp 4	**Procedures & Control Processes**	Trách nhiệm, trình tự, điểm kiểm soát và bằng chứng.
Cấp 5	**Guidelines, Playbooks & Templates**	Hướng dẫn thực hành, checklist, ví dụ và biểu mẫu.

Tài liệu cấp dưới có thể được cập nhật thường xuyên để thích ứng với công nghệ, rủi ro và mô hình hoạt động, nhưng không được mâu thuẫn với Hiến chương này.

5. Mười nguyên tắc nền tảng

TEN FOUNDATIONAL PRINCIPLES

Điều 1. Ưu tiên giá trị kinh doanh

Business Value First

Chúng ta phát triển công nghệ để giải quyết vấn đề thực tế, phục vụ người dùng và tạo ra kết quả có ý nghĩa cho tổ chức.

1. Mỗi sáng kiến kỹ thuật phải liên kết với nhu cầu kinh doanh, nhu cầu người dùng, nghĩa vụ pháp lý hoặc mục tiêu giảm thiểu rủi ro được xác định.
2. Phải hiểu người dùng, vấn đề, giá trị mong đợi, tiêu chí thành công, chi phí và rủi ro trước khi cam kết phát triển.
3. Yêu cầu phải thể hiện mục đích và kết quả mong đợi, không chỉ mô tả chức năng cần xây dựng.
4. Khi giá trị chưa được kiểm chứng, ưu tiên thử nghiệm nhỏ, prototype hoặc MVP trước khi đầu tư quy mô lớn.
5. Khi tính năng không còn tạo giá trị tương xứng, tổ chức phải sẵn sàng đơn giản hóa, thay thế hoặc ngừng sử dụng.
6. Không sử dụng danh nghĩa giá trị kinh doanh để bỏ qua bảo mật, quyền riêng tư, chất lượng hoặc tuân thủ.

NGUYÊN TẮC ƯU TIÊN *Kết quả kinh doanh và giá trị người dùng cao hơn sản lượng hoạt động.*

Điều 2. Bảo mật và quyền riêng tư từ khâu thiết kế

Security and Privacy by Design

Bảo mật và quyền riêng tư là thuộc tính cốt lõi của hệ thống, không phải thành phần được bổ sung sau khi phát triển hoàn tất.

1. Yêu cầu bảo mật, quyền riêng tư và tuân thủ phải được xác định từ giai đoạn phân tích và thiết kế.
2. Thiết kế phải xem xét mối đe dọa, tình huống lạm dụng, điểm tấn công và biện pháp giảm thiểu.
3. Quyền truy cập tuân theo least privilege, need to know và default deny.
4. Dữ liệu phải được phân loại, bảo vệ, lưu giữ và tiêu hủy tương xứng với mức độ nhạy cảm.
5. Credentials, access keys, personal data và sensitive data không được lưu trực tiếp trong mã nguồn hoặc log không được bảo vệ.
6. Thành phần mã nguồn mở, thư viện và dịch vụ bên ngoài phải được quản trị như chuỗi cung ứng phần mềm.
7. Security testing phải được tích hợp phù hợp vào SDLC và CI/CD.
8. Chấp nhận rủi ro bảo mật phải do người có thẩm quyền thực hiện.

NGUYÊN TẮC ƯU TIÊN *Bảo vệ khách hàng, dữ liệu và tổ chức cao hơn sự thuận tiện trong phát triển hoặc vận hành.*

Điều 3. Chất lượng là trách nhiệm của tất cả mọi người

Quality Is Everyone's Responsibility

Mỗi cá nhân tham gia vào vòng đời sản phẩm đều chịu trách nhiệm đối với chất lượng kết quả cuối cùng.

1. Product và Business Analysis chịu trách nhiệm về giá trị, tính rõ ràng và khả năng kiểm chứng của yêu cầu.
2. Architect chịu trách nhiệm xem xét các thuộc tính chất lượng và hệ quả dài hạn của thiết kế.
3. Engineer chịu trách nhiệm đối với chất lượng mã nguồn, kiểm thử kỹ thuật và khả năng bảo trì.
4. Quality Engineer cung cấp góc nhìn độc lập và nâng cao năng lực bảo đảm chất lượng của toàn nhóm.
5. DevOps, Platform và Operations chịu trách nhiệm về khả năng triển khai, vận hành, giám sát và phục hồi.
6. Chất lượng phải được xây dựng trong từng giai đoạn, ưu tiên phòng ngừa và phát hiện sớm.
7. Rủi ro chất lượng đã biết phải được công khai và chấp thuận phù hợp trước khi phát hành.

NGUYÊN TẮC ƯU TIÊN *Chất lượng được tạo ra bởi cả hệ thống làm việc, không phải bởi một cổng kiểm tra cuối cùng.*

Điều 4. Phân tách trách nhiệm

Segregation of Duties

Không cá nhân nào được toàn quyền thực hiện, phê duyệt và che giấu một thay đổi quan trọng do chính mình tạo ra.

1. Người tạo ra thay đổi quan trọng không được là người duy nhất phê duyệt thay đổi đó.
2. Mã nguồn quan trọng phải được peer review trước khi hợp nhất hoặc phát hành.
3. Quyền phát triển, phê duyệt và triển khai production phải được phân tách theo mức độ rủi ro.
4. Quyền truy cập đặc quyền phải được hạn chế, phê duyệt, ghi nhận và rà soát định kỳ.
5. Khi không thể phân tách đầy đủ, phải có control thay thế như independent review, audit log, time-bound access hoặc management review.
6. Quy trình khẩn cấp có thể rút gọn nhưng không loại bỏ khả năng truy vết và kiểm tra độc lập.

NGUYÊN TẮC ƯU TIÊN *Trao quyền phải đi kèm kiểm soát độc lập và khả năng truy vết.*

Điều 5. Mọi thứ đều là code khi khả thi

Everything as Code Where Practical

Hạ tầng, cấu hình, kiểm thử, policy và quy trình triển khai nên được định nghĩa bằng code khi khả thi, an toàn và tạo ra giá trị lâu dài.

1. Cấu hình và hạ tầng quan trọng nên được quản lý bằng version control.
2. Tạo mới và thay đổi môi trường nên được tự động hóa để bảo đảm tính nhất quán và khả năng lặp lại.
3. Build, test, security scanning và deployment nên được tích hợp vào pipeline tự động khi khả thi.
4. Automation code phải được review, kiểm thử và bảo vệ tương xứng với rủi ro.
5. Thay đổi trực tiếp trên production phải được hạn chế, ghi nhận và đưa trở lại code khi phù hợp.
6. Không tự động hóa quy trình chưa được hiểu rõ hoặc khi chi phí duy trì vượt xa giá trị.
7. Trường hợp chưa thể thực hiện dưới dạng code phải có tài liệu, control và ownership rõ ràng.

NGUYÊN TẮC ƯU TIÊN *Khả năng lặp lại, kiểm chứng và truy vết cao hơn thao tác thủ công phụ thuộc vào cá nhân.*

Điều 6. Có thể quan sát và vận hành được ngay từ đầu

Observable and Operable by Default

Một hệ thống chưa sẵn sàng nếu đội ngũ không thể biết nó đang hoạt động thế nào, phát hiện bất thường hoặc điều tra sự cố.

1. Logging, monitoring, tracing và alerting phải được xem xét trong thiết kế.
2. Hệ thống phải cung cấp đủ tín hiệu về availability, chức năng, hiệu năng, lỗi, phạm vi ảnh hưởng và tính chính xác của xử lý dữ liệu.
3. Log nên có cấu trúc, timestamp, correlation identifier và ngữ cảnh phù hợp.
4. Log không được làm lộ credentials, tokens, personal data hoặc dữ liệu nhạy cảm không cần thiết.
5. Alert phải có ý nghĩa, có owner và dẫn đến hành động rõ ràng.
6. Hệ thống quan trọng phải có runbook; dependency, backup, recovery và rollback phải được xác định trước production.
7. Khả năng hỗ trợ và vận hành là một phần của tiêu chí hoàn thành.

NGUYÊN TẮC ƯU TIÊN *Một hệ thống có thể quan sát và phục hồi được tốt hơn một hệ thống phức tạp nhưng không thể chẩn đoán.*

Điều 7. Thay đổi nhỏ và có thể đảo ngược

Small, Reversible Changes

Chúng ta ưu tiên các thay đổi nhỏ, độc lập, có thể kiểm tra, triển khai và đảo ngược an toàn.

1. Thay đổi lớn nên được chia thành các thay đổi nhỏ hơn khi khả thi.
2. Mỗi thay đổi phải có phạm vi, mục đích, owner và mức kiểm thử tương xứng với rủi ro.
3. Triển khai nên hỗ trợ rollback hoặc roll-forward an toàn.
4. Feature flags, canary, blue-green hoặc progressive rollout nên được dùng khi phù hợp.
5. Database changes phải xem xét backward compatibility, tính toàn vẹn dữ liệu và khả năng phục hồi.
6. Thay đổi phải có cơ chế xác nhận kết quả sau triển khai.
7. Thay đổi không thể đảo ngược phải được công khai, đánh giá rủi ro và có kế hoạch phục hồi.
8. Thay đổi khẩn cấp vẫn phải được ghi nhận, giám sát và đánh giá sau triển khai.

NGUYÊN TẮC ƯU TIÊN *Thay đổi nhỏ, thường xuyên và có thể khôi phục tốt hơn thay đổi lớn, hiếm và khó kiểm soát.*

Điều 8. Bằng chứng quan trọng hơn ý kiến cá nhân

Evidence over Opinion

Quyết định kỹ thuật phải dựa trên dữ liệu, thử nghiệm, yêu cầu, rủi ro và bằng chứng có thể kiểm chứng.

1. Lựa chọn kỹ thuật phải được đánh giá dựa trên tiêu chí rõ ràng.
2. Khi có bất đồng, ưu tiên dữ liệu thực tế, thử nghiệm, prototype, benchmark phù hợp, phân tích rủi ro, phản hồi người dùng và kết quả vận hành.
3. Quyết định kiến trúc quan trọng phải được ghi lại bằng ADR hoặc hình thức tương đương.
4. ADR phải nêu bối cảnh, phương án, quyết định, lý do, hệ quả và điều kiện đánh giá lại.
5. Khi chưa đủ bằng chứng, phải công khai giả định và mức độ không chắc chắn.
6. Ý kiến chuyên gia có giá trị nhưng phải được nhận diện là nhận định chuyên môn.
7. Quyết định được phép thay đổi khi xuất hiện bằng chứng mới đáng tin cậy.

NGUYÊN TẮC ƯU TIÊN *Quyết định có thể giải thích và kiểm chứng tốt hơn quyết định dựa trên thẩm quyền hoặc sở thích.*

Điều 9. Tái sử dụng trước khi xây mới

Reuse Before Build

Trước khi xây dựng một giải pháp mới, chúng ta phải kiểm tra có hệ thống khả năng tái sử dụng, mở rộng, mua hoặc tích hợp năng lực hiện có.

1. Nhóm đề xuất phải tìm hiểu các sản phẩm, shared platform, API, dữ liệu và dịch vụ hiện có.
2. Không xây dựng trùng lặp một năng lực đã tồn tại nếu không có lý do chính đáng.
3. Quyết định reuse, buy, extend hoặc build phải xem xét tổng chi phí sở hữu.
4. Tái sử dụng phải đánh giá functional fit, security, scalability, supportability, licensing và vendor risk.
5. Không tái sử dụng giải pháp không phù hợp chỉ để tránh xây mới.
6. Shared components phải có owner, versioning, documentation và support model rõ ràng.
7. Khi xây mới, phải ghi nhận lý do các lựa chọn tái sử dụng không phù hợp.

NGUYÊN TẮC ƯU TIÊN *Tái sử dụng có trách nhiệm tốt hơn xây dựng trùng lặp; phù hợp với mục đích tốt hơn tái sử dụng bằng mọi giá.*

Điều 10. Học hỏi liên tục

Continuous Learning

Mỗi dự án, thay đổi, sự cố, retrospective và phản hồi người dùng phải trở thành cơ hội cải tiến.

1. Các nhóm phải định kỳ phản ánh về hiệu quả làm việc và xác định hành động cải tiến cụ thể.
2. Sự cố nghiêm trọng phải được đánh giá về diễn biến, ảnh hưởng, nguyên nhân hệ thống, khả năng phát hiện và biện pháp phòng ngừa.
3. Post-incident review tập trung vào hệ thống, điều kiện và control, không tìm người để đổ lỗi.
4. Blameless culture không miễn trừ hành vi cố ý, che giấu, thiếu trung thực hoặc coi thường rủi ro.
5. Mỗi retrospective hoặc review phải tạo ra hành động có owner và thời hạn, chấp nhận rủi ro có chủ đích, bài học hoặc quyết định không hành động kèm lý do.
6. Bài học có giá trị phải được chia sẻ và hành động cải tiến phải được theo dõi đến khi đóng.
7. Standards và cách làm phải được điều chỉnh khi bằng chứng cho thấy không còn hiệu quả.

NGUYÊN TẮC ƯU TIÊN *Sai sót được chuyển hóa thành cải tiến tốt hơn sai sót bị che giấu hoặc lặp lại.*

6. Nguyên tắc giải quyết xung đột

CONFLICT RESOLUTION

Mười nguyên tắc phải được xem xét như một hệ thống thống nhất. Không được áp dụng một nguyên tắc riêng lẻ theo cách làm mất hiệu lực của các nguyên tắc còn lại.

- Business Value First không cho phép bỏ qua bảo mật hoặc chất lượng.
- Reuse Before Build không bắt buộc sử dụng một giải pháp không phù hợp.
- Everything as Code không bắt buộc tự động hóa khi chi phí vượt quá giá trị.
- Small, Reversible Changes không có nghĩa mọi thay đổi đều có thể rollback tức thời.
- Evidence over Opinion không loại bỏ vai trò kinh nghiệm chuyên môn khi dữ liệu chưa đầy đủ.
- Continuous Learning không được sử dụng để miễn trừ trách nhiệm.

Khi mục tiêu cạnh tranh, thứ tự ưu tiên mặc định là:

1. An toàn con người và tuân thủ pháp luật.
2. Bảo mật, quyền riêng tư và bảo vệ dữ liệu.
3. Tính toàn vẹn của giao dịch và thông tin.
4. Lợi ích và sự tin cậy của khách hàng.
5. Khả năng vận hành và phục hồi dịch vụ.
6. Giá trị kinh doanh.
7. Chất lượng và tính bền vững kỹ thuật.
8. Tốc độ cung cấp.
9. Tối ưu chi phí ngắn hạn.

Việc áp dụng phải dựa trên mức độ quan trọng, ảnh hưởng và rủi ro của từng hệ thống. Không phải mọi hệ thống đều cần cùng một mức kiểm soát.

7. Quyền tự chủ và trách nhiệm của nhóm kỹ thuật

AUTONOMY AND ACCOUNTABILITY

Các nhóm kỹ thuật được trao quyền để:

- Lựa chọn cách thức thực hiện trong phạm vi guardrails đã được phê duyệt.
- Đề xuất công nghệ, thiết kế và phương pháp làm việc.
- Thực hiện prototype và thử nghiệm có kiểm soát.
- Yêu cầu làm rõ giá trị kinh doanh và tiêu chí thành công.
- Thách thức quyết định khi có bằng chứng hợp lý.
- Từ chối triển khai khi có rủi ro nghiêm trọng chưa được xử lý.
- Nêu quan ngại kỹ thuật, bảo mật hoặc đạo đức mà không sợ bị trả đũa.

Các nhóm đồng thời có trách nhiệm:

- Tuân thủ Hiến chương và các yêu cầu bắt buộc.
- Công khai rủi ro, giới hạn và nợ kỹ thuật.
- Cung cấp bằng chứng về chất lượng và mức độ sẵn sàng.
- Bảo vệ dữ liệu, tài sản và uy tín của tổ chức.
- Bảo đảm giải pháp có thể vận hành và hỗ trợ.
- Hợp tác, học hỏi và cải tiến liên tục.
- Không sử dụng quyền tự chủ để bỏ qua kiểm soát hoặc trách nhiệm giải trình.

8. Ngoại lệ

EXCEPTIONS

Ngoại lệ là cơ chế quản trị có kiểm soát, không phải cách bỏ qua Hiến chương hoặc tiêu chuẩn. Mỗi yêu cầu ngoại lệ phải bao gồm:

1. Nguyên tắc, policy hoặc standard chưa thể đáp ứng.
2. Lý do kinh doanh hoặc kỹ thuật.
3. Phạm vi và hệ thống bị ảnh hưởng.
4. Thời gian áp dụng.
5. Rủi ro phát sinh và control thay thế.
6. Người chịu trách nhiệm xử lý.
7. Người có thẩm quyền chấp nhận rủi ro.
8. Ngày hết hiệu lực.
9. Kế hoạch khắc phục hoặc đánh giá lại.

Không được cấp ngoại lệ đối với yêu cầu pháp luật, hành vi gian lận hoặc che giấu, xử lý dữ liệu trái phép, tự phê duyệt thay đổi quan trọng mà không có control thay thế, hoặc rủi ro nghiêm trọng không có biện pháp giảm thiểu phù hợp.

9. Quản trị và trách nhiệm thực thi

GOVERNANCE AND ENFORCEMENT

Technology Governance Council

- Bảo vệ tính toàn vẹn của Hiến chương.
- Phê duyệt Engineering Policies trọng yếu.
- Giải quyết xung đột trong diễn giải nguyên tắc.
- Phê duyệt ngoại lệ rủi ro cao và đề xuất sửa đổi Hiến chương.

Engineering Leadership

- Chuyển nguyên tắc thành policies, standards và guardrails.
- Cung cấp nền tảng và công cụ giúp tuân thủ.
- Bảo đảm yêu cầu quản trị có khả năng triển khai.
- Thiết lập đo lường và xử lý rào cản hệ thống.

Engineering Teams

- Áp dụng Hiến chương trong hoạt động hằng ngày.
- Cung cấp bằng chứng tuân thủ và báo cáo ngoại lệ.
- Phản ánh quy định không còn phù hợp.
- Chuyển hóa bài học thành cải tiến.

Architecture, Security, Risk & Compliance

- Chuyển nghĩa vụ quản trị thành yêu cầu rõ ràng.
- Thiết kế control tương xứng với rủi ro và duy trì tính độc lập.
- Phối hợp để tránh control không khả thi.
- Đánh giá dựa trên bằng chứng.

10. Đánh giá việc áp dụng

ASSESSMENT

Việc áp dụng Hiến chương không được đánh giá chỉ bằng số lượng tài liệu, cuộc họp hoặc approvals. Bằng chứng có thể bao gồm:

- Mức độ liên kết giữa sản phẩm và kết quả kinh doanh.
- Phản hồi và mức độ sử dụng của người dùng.
- Kết quả security assessment và chất lượng kiểm thử.
- Tỷ lệ thay đổi được peer review và mức độ phân tách trách nhiệm.
- Tỷ lệ hạ tầng, cấu hình và pipeline được quản lý bằng code.
- Mức độ bao phủ logging, monitoring và alerting.
- Tần suất triển khai, tỷ lệ thay đổi thành công và khả năng recovery.
- Mức độ tái sử dụng và chất lượng ADR.
- Tình trạng hoàn tất hành động từ retrospective và incident review.
- Mức độ phụ thuộc vào thao tác thủ công hoặc cá nhân chủ chốt.

Các chỉ số được sử dụng để học hỏi và cải tiến. Không sử dụng một chỉ số đơn lẻ làm mục tiêu tuyệt đối nếu điều đó tạo ra hành vi đối phó hoặc làm sai lệch kết quả.

11. Sửa đổi Hiến chương

AMENDMENT

Hiến chương được thiết kế để ổn định trong nhiều năm và chỉ nên sửa đổi khi chiến lược, mô hình hoạt động, yêu cầu pháp lý hoặc loại rủi ro trọng yếu thay đổi, hoặc khi bằng chứng thực tế chứng minh nguyên tắc không còn phù hợp.

1. Nêu rõ vấn đề cần giải quyết.
2. Phân tích ảnh hưởng đến policies và standards hiện hành.
3. Tham vấn Engineering, Architecture, Security, Risk, Operations và đại diện Business.
4. Được Technology Governance Council xem xét.
5. Được [CTO/CIO/Executive Committee] phê duyệt.
6. Có kế hoạch truyền thông nội bộ và chuyển tiếp.

Thay đổi công cụ, nền tảng cloud, framework, ngôn ngữ lập trình hoặc phương pháp triển khai thông thường không phải là lý do để sửa đổi Hiến chương.

12. Cam kết kỹ thuật

ENGINEERING COMMITMENT

Hiểu rõ giá trị trước khi xây dựng;
bảo vệ khách hàng và dữ liệu ngay từ thiết kế;
cùng chịu trách nhiệm về chất lượng;
duy trì sự độc lập trong kiểm soát;
tự động hóa những gì hợp lý;
xây dựng hệ thống có thể quan sát và vận hành;
thực hiện các thay đổi nhỏ và có thể phục hồi;
đưa ra quyết định dựa trên bằng chứng;
tái sử dụng có trách nhiệm trước khi xây mới;
và chuyển hóa mỗi kinh nghiệm thành sự cải tiến.

13. Phê duyệt

APPROVAL

Vai trò	Họ tên	Chữ ký	Ngày
Executive Sponsor			
Chief Technology Officer			
Head of Technology Center			
Head of Technology Risk			

Phụ lục A. Gợi ý hệ thống tài liệu triển khai

SUGGESTED SUPPORTING DOCUMENTS

Policies

Software Development; Security; Change & Release; Cloud; Open Source; Data; AI; Production Management.

Standards

Coding; Secure Coding; API; Testing; Logging & Monitoring; CI/CD; Infrastructure as Code; ADR.

Procedures

Code Review; Production Deployment; Emergency Change; Vulnerability Remediation; Architecture Review; Incident Management.

Guidelines & Playbooks

Technology selection; reusable components; operational readiness; incident response; retrospective; exception template.

NGUYÊN TẮC ƯU TIÊN *Các ngưỡng đo lường, công cụ, công nghệ, thời gian xử lý và chi tiết control nên nằm trong Policy hoặc Standard, không đưa vào Hiến chương.*