SOFTWARE QUALITY POLICY

CHÍNH SÁCH CHẤT LƯỢNG PHẦN MỀM

Built-in Quality • Shift Left Testing • Measure, Don't Guess

Tổ chức / Organization	[Tên tổ chức]
Chủ sở hữu / Document Owner	Head of Quality / QA Lead
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 có thay đổi đáng kể về công cụ/mô hình chất lượng tham chiếu
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 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. Quy định CỤ THỂ các yêu cầu bắt buộc về chất lượng phần mềm mà Software Development Governance Policy (SDGP) mới chỉ nêu ở mức nguyên tắc chung tại Điều 5.3 ("Chất lượng trong từng giai đoạn"), nhằm bảo đảm chất lượng được xây dựng ngay trong quá trình phát triển (built-in quality) thay vì chỉ được kiểm tra ở cuối vòng đời.
2. Thiết lập một bộ tiêu chí, ngưỡng và cơ chế đo lường nhất quán để "chất lượng" không còn là khái niệm cảm tính, mà là điều có thể kiểm tra, tự động hóa và đối chiếu bằng dữ liệu (measure, don't guess).
3. Bảo đảm nợ kỹ thuật, rủi ro chất lượng đã biết và lỗi phát hiện sau phát hành được quản lý chủ động thay vì bị che giấu hoặc tích lũy không kiểm soát.

Chính sách này bổ sung cho SDGP, không thay thế — các nội dung về SDLC, vai trò/trách nhiệm chung, cơ chế ngoại lệ và không tuân thủ đã được quy định tại SDGP và áp dụng nguyên vẹn cho chính sách này (xem Điều 8, Điều 9).

2. Phạm vi áp dụng

SCOPE

- Áp dụng cho toàn bộ phần mềm nội bộ, phần mềm phục vụ khách hàng và các thành phần hạ tầng dạng code (Infrastructure as Code) do tổ chức phát triển hoặc phát triển thuê ngoài.
- Áp dụng cho mọi giai đoạn trong SDLC nêu tại SDGP Điều 5.1, từ thiết kế, viết mã, kiểm thử, đến vận hành sau phát hành.
- Áp dụng cho toàn bộ đội phát triển, QA/kiểm thử, DevOps/nền tảng và bất kỳ vai trò nào tham gia tạo ra hoặc kiểm soát chất lượng phần mềm.
- Không quy định lại yêu cầu bảo mật trong code review (thuộc Secure Software Development Policy) hoặc yêu cầu bảo vệ dữ liệu test (đã quy định tại SDGP Điều 5.2 về phân tách môi trường và ẩn danh hóa dữ liệu). Chính sách này tập trung vào khía cạnh chất lượng: kiểm thử, khả năng đọc hiểu/bảo trì mã nguồn, và quản lý nợ kỹ thuật.

3. Liên kết với Engineering Constitution

ALIGNMENT WITH THE CONSTITUTION

Yêu cầu trong Policy	Nguyên tắc Constitution tương ứng
5.1 Definition of Done tối thiểu bắt buộc	Điều 3 — Quality Is Everyone's Responsibility
5.2 Chiến lược kiểm thử theo test pyramid	Điều 3, Điều 8 — Quality / Evidence over Opinion
5.3 Quality gate tự động trong CI/CD	Điều 5 — Everything as Code Where Practical
5.4 Quản lý nợ kỹ thuật	Điều 7 — Small, Reversible Changes
5.5 Đo lường defect escape rate và phản hồi ngược	Điều 8, Điều 10 — Evidence over Opinion / Continuous Learning

5.6 Code review chất lượng (đọc hiểu, bảo trì, coding standard)	Điều 3, Điều 4 — Quality / Segregation of Duties
5.7 Công khai rủi ro chất lượng chưa xử lý	Điều 8 — Evidence over Opinion

4. Định nghĩa

DEFINITIONS

- **Mô hình chất lượng phần mềm (ISO/IEC 25010):** khung tham chiếu quốc tế mô tả chất lượng phần mềm qua các đặc tính có thể đánh giá — functional suitability (phù hợp chức năng), performance efficiency (hiệu năng), compatibility (khả năng tương thích), usability/interaction capability (khả năng sử dụng/tương tác), reliability (độ tin cậy), security (bảo mật), maintainability (khả năng bảo trì), portability/flexibility (khả năng di động/linh hoạt), và safety (an toàn — đặc tính bổ sung trong bản cập nhật 2023). Chính sách này dùng mô hình này làm khung tham chiếu khi đánh giá chất lượng, không giới hạn "chất lượng" chỉ ở việc "không có lỗi chức năng".
- **Test Pyramid:** mô hình chiến lược kiểm thử tự động do Mike Cohn đề xuất, khuyến nghị tỷ trọng giảm dần từ đáy lên đỉnh: nhiều unit test (nhanh, rẻ, cô lập), một lượng vừa phải integration test (kiểm tra tương tác giữa các thành phần), và ít end-to-end/UI test (chậm, đắt, dễ vỡ) ở trên cùng.
- **Flaky Test (Test chập chờn):** test case cho kết quả không nhất quán (lúc pass, lúc fail) giữa các lần chạy dù mã nguồn không thay đổi, làm giảm độ tin cậy của tín hiệu từ pipeline CI/CD.
- **Definition of Done (DoD):** tập tiêu chí bắt buộc để một hạng mục công việc được coi là hoàn thành, đã được định nghĩa khung tại SDGP Điều 4 (tối thiểu: đã kiểm thử, đã review, đáp ứng Coding/Secure Coding Standard, có tài liệu cần thiết). Điều 5.1 của chính sách này quy định chi tiết nội dung tối thiểu bắt buộc của DoD.
- **Quality Gate:** điểm kiểm soát tự động trong pipeline CI/CD, tại đó một thay đổi chỉ được phép đi tiếp (merge/deploy) nếu đáp ứng ngưỡng chất lượng đã quy định trước.
- **Nợ kỹ thuật (Technical Debt):** hệ quả của các lựa chọn thiết kế/triển khai đơn giản hóa hoặc chưa tối ưu, được chấp nhận có chủ đích hoặc phát sinh ngoài ý muốn, cần được xử lý dần để tránh làm chậm tốc độ phát triển hoặc tăng rủi ro chất lượng về sau.
- **Defect Escape Rate:** tỷ lệ hoặc số lượng lỗi được phát hiện sau khi phần mềm đã phát hành (production) thay vì được phát hiện trong các giai đoạn kiểm thử trước đó — chỉ số phản ánh hiệu quả thực tế của quy trình kiểm thử nội bộ.
- **DORA/Accelerate Metrics:** bốn chỉ số hiệu năng phân phối phần mềm do nghiên cứu DORA (DevOps Research and Assessment) xác lập — deployment frequency (tần suất triển khai), lead time for changes (thời gian từ commit đến production), change failure rate (tỷ lệ triển khai gây lỗi), và mean time to restore/MTTR (thời gian trung bình khôi phục dịch vụ sau sự cố; từ năm 2023 DORA thu hẹp định nghĩa này thành "failed deployment recovery time" để chỉ tính thời gian khôi phục do chính thay đổi gây ra, tách biệt khỏi sự cố hạ tầng bên ngoài). SDGP Điều 7 đã dùng nhóm chỉ số này ở mục "Delivery"; chính sách này bổ sung chỉ số "Quality" đi kèm.

5. Yêu cầu bắt buộc

MANDATORY REQUIREMENTS

5.1 Definition of Done tối thiểu bắt buộc

1. Mọi đội phát triển **PHẢI (MUST)** áp dụng một Definition of Done bằng văn bản cho mọi hạng mục công việc (user story, task, bug fix), với nội dung tối thiểu không được thấp hơn: đã vượt qua kiểm thử tự động ở phạm vi tương ứng theo Điều 5.2; đã được ít nhất một người review độc lập theo Điều 5.6 và theo SDGP Điều 5.4; đáp ứng Coding Standard (và Secure Coding Standard khi áp dụng); đã đáp ứng các tiêu chí chấp nhận (acceptance criteria) được thống nhất từ giai đoạn thiết kế/lấy yêu cầu; và có tài liệu cần thiết đi kèm (cập nhật API doc, README, release note, hoặc ghi chú vận hành nếu có thay đổi hành vi hệ thống).
2. Đội phát triển **ĐƯỢC PHÉP (MAY)** bổ sung tiêu chí DoD chặt hơn mức tối thiểu này tùy theo mức độ rủi ro của hệ thống, nhưng **KHÔNG ĐƯỢC PHÉP (MUST NOT)** hạ thấp hơn mức tối thiểu tại Điều 5.1.1 nếu không xin ngoại lệ theo Điều 8.
3. DoD của mỗi đội **PHẢI (MUST)** được công khai trong công cụ quản lý backlog và rà soát lại khi có thay đổi đáng kể về công cụ hoặc rủi ro hệ thống.

5.2 Chiến lược kiểm thử theo test pyramid

1. Chiến lược kiểm thử tự động của mỗi hệ thống **PHẢI (MUST)** được thiết kế theo nguyên tắc test pyramid tại Điều 4: tỷ trọng lớn nhất dành cho unit test, tỷ trọng vừa phải cho integration test, và tỷ trọng nhỏ nhất cho end-to-end/UI test — nhằm giữ vòng phản hồi (feedback loop) nhanh và chi phí bảo trì bộ test ở mức chấp nhận được.
2. Ngoài kiểm thử chức năng, chiến lược kiểm thử **PHẢI (MUST)** bao gồm kiểm thử phi chức năng (hiệu năng, khả năng truy cập, bảo mật cơ bản...) tương xứng với rủi ro và đặc thù của từng hệ thống, đối chiếu các đặc tính chất lượng liên quan tại mô hình ISO/IEC 25010 nêu ở Điều 4.
3. Ngưỡng bao phủ kiểm thử tự động (test coverage) **PHẢI (MUST)** được quy định tương xứng với mức độ rủi ro của từng hệ thống hoặc thành phần (ví dụ: thành phần xử lý thanh toán, dữ liệu cá nhân hoặc logic nghiệp vụ cốt lõi cần ngưỡng cao hơn thành phần giao diện ít rủi ro), theo chi tiết tại Testing Standard (Tầng 3).
4. Tổ chức **KHÔNG ĐƯỢC PHÉP (MUST NOT)** áp đặt một ngưỡng coverage đồng nhất tuyệt đối (ví dụ 100%) cho mọi hệ thống không phân biệt rủi ro — yêu cầu như vậy thường không khả thi, tạo động lực viết test hình thức (test không có khả năng phát hiện lỗi) chỉ để đạt chỉ số, và không phản ánh đúng mục tiêu chất lượng thực chất. Coverage tự động **PHẢI (MUST)** được xem là một chỉ báo hỗ trợ, không phải mục tiêu tự thân.
5. Test case chập chờn (flaky test, theo định nghĩa tại Điều 4) **PHẢI (MUST)** được đưa vào diện cách ly (quarantine) khỏi luồng chặn merge/deploy chính ngay khi phát hiện, kèm ticket theo dõi xử lý; test đang cách ly **KHÔNG ĐƯỢC PHÉP (MUST NOT)** được tính vào kết quả coverage cho đến khi được khắc phục và xác nhận ổn định trở lại.

5.3 Quality gate tự động trong CI/CD

1. Pipeline CI/CD của hệ thống **PHẢI (MUST)** có ít nhất một quality gate tự động chặn merge vào nhánh dùng chung hoặc chặn deploy production khi thay đổi không đáp ứng ngưỡng chất lượng tối thiểu đã

quy định (kết quả kiểm thử tự động bắt buộc, ngưỡng coverage theo Điều 5.2.3, kết quả phân tích tĩnh mã nguồn).

2. Việc bỏ qua (override) một quality gate tự động **PHẢI (MUST)** được ghi log, nêu rõ lý do và được một người có thẩm quyền không phải tác giả thay đổi phê duyệt — áp dụng theo nguyên tắc phân tách trách nhiệm tại SDGP Điều 5.4.
3. Yêu cầu kỹ thuật cụ thể của từng loại quality gate (công cụ, ngưỡng số liệu, cơ chế cách ly flaky test) do CI/CD Standard và Testing Standard (Tầng 3) quy định.

5.4 Quản lý nợ kỹ thuật

1. Nợ kỹ thuật được phát hiện hoặc chấp nhận có chủ đích (ví dụ để kịp thời hạn phát hành) **PHẢI (MUST)** được ghi nhận bằng văn bản vào backlog hoặc sổ theo dõi nợ kỹ thuật, kèm mô tả rủi ro và mức độ ưu tiên xử lý — không được để tồn tại "ngầm" chỉ trong hiểu biết cá nhân của một kỹ sư.
2. Mỗi đội phát triển, cùng với Product Owner/đại diện nghiệp vụ, **PHẢI (MUST)** thống nhất phân bổ một tỷ lệ dung lượng (capacity) định kỳ theo mỗi chu kỳ phát triển để xử lý nợ kỹ thuật đã ghi nhận — thực tiễn ngành thường tham khảo mức khoảng 15-20% mỗi chu kỳ lặp làm điểm khởi đầu, được điều chỉnh tăng/giảm tùy mức độ nợ tồn đọng và không được xem là con số cố định tuyệt đối cho mọi đội.
3. Nợ kỹ thuật có nguy cơ ảnh hưởng trực tiếp đến bảo mật, độ tin cậy hoặc khả năng mở rộng của hệ thống **PHẢI (MUST)** được đưa vào rủi ro chất lượng công khai theo Điều 5.7, không chờ đến kỳ phân bổ dung lượng định kỳ nếu mức độ nghiêm trọng cao.
4. Nợ kỹ thuật tồn đọng quá lâu không xử lý **NÊN (SHOULD)** được báo cáo định kỳ cho Engineering Governance Council như một phần của đo lường tuân thủ tại Điều 7.

5.5 Đo lường defect escape rate và phản hồi ngược

1. Mỗi hệ thống **PHẢI (MUST)** đo lường defect escape rate (Điều 4) theo chu kỳ định kỳ, làm chỉ số "Quality" bổ sung cho nhóm chỉ số "Delivery" (DORA metrics) đã nêu tại SDGP Điều 7.
2. Khi defect escape rate vượt ngưỡng cảnh báo do Engineering Leadership quy định, đội phát triển liên quan **PHẢI (MUST)** thực hiện rà soát nguyên nhân gốc rễ (root cause) và xác định thay đổi cụ thể trong chiến lược kiểm thử (Điều 5.2) hoặc quy trình review (Điều 5.6) để giảm tái diễn — theo đúng tinh thần học hỏi liên tục tại SDGP Điều 5.13.
3. Kết quả rà soát và thay đổi quy trình rút ra **NÊN (SHOULD)** được chia sẻ giữa các đội để tránh lặp lại cùng một loại lỗi ở hệ thống khác.

5.6 Code review chất lượng

1. Ngoài các yêu cầu bảo mật đã quy định tại Secure Software Development Policy, người review **PHẢI (MUST)** đánh giá thêm tối thiểu: khả năng đọc hiểu của mã nguồn (đặt tên rõ ràng, cấu trúc logic mạch lạc), khả năng bảo trì (tránh trùng lặp không cần thiết, mức độ phức tạp hợp lý), và mức độ tuân thủ Coding Standard đã ban hành.
2. Đối với mã nguồn hoặc test case có sự đóng góp đáng kể từ công cụ AI, người review **PHẢI (MUST)** thẩm định thêm tính hợp lý của logic và điều kiện kiểm tra (assertion) — không chấp nhận test case chỉ nhằm tăng chỉ số coverage mà không có khả năng phát hiện lỗi thực tế; yêu cầu chung về dùng AI hỗ trợ phát triển do AI-Assisted Software Development Policy quy định.
3. Người review **KHÔNG ĐƯỢC PHÉP (MUST NOT)** chỉ dựa vào việc pipeline CI đã "xanh" (pass) để phê duyệt thay đổi mà bỏ qua việc đọc hiểu nội dung thay đổi.

4. Góp ý review liên quan đến khả năng đọc hiểu/bảo trì **NÊN (SHOULD)** được đưa ra mang tính xây dựng và có thể tham chiếu đến điều khoản cụ thể trong Coding Standard, tránh ý kiến chủ quan không có cơ sở.

5.7 Công khai rủi ro chất lượng chưa xử lý

1. Kế thừa nguyên tắc tại SDGP Điều 5.3, mọi rủi ro chất lượng đã biết nhưng chưa được xử lý trước một lần phát hành **PHẢI (MUST)** được ghi nhận vào một sổ theo dõi rủi ro chất lượng (quality risk log), nêu rõ: mô tả rủi ro, mức độ ảnh hưởng ước tính, biện pháp giảm thiểu tạm thời (nếu có), và kế hoạch xử lý dứt điểm.
2. Việc phát hành khi còn rủi ro chất lượng chưa xử lý trong risk log **PHẢI (MUST)** được một người có thẩm quyền (không phải tác giả thay đổi) xác nhận chấp nhận rủi ro bằng văn bản (sign-off) trước khi triển khai production.
3. Risk log **PHẢI (MUST)** được lưu trữ và có thể truy xuất phục vụ audit hoặc điều tra sự cố sau này.

> **Không lặp lại nội dung của SDGP:** Các yêu cầu về SDLC, vai trò/trách nhiệm chung, phân tách môi trường/dữ liệu, quản lý thay đổi, cơ chế ngoại lệ và không tuân thủ đã được quy định đầy đủ tại SDGP. Policy này chỉ mở rộng sâu phần chất lượng (Điều 5.3 của SDGP); khi có xung đột hoặc trùng lặp diễn giải, bản quy định tại SDGP được ưu tiên áp dụng cho phần chung.

6. Vai trò và trách nhiệm

ROLES & RESPONSIBILITIES

- **Engineering Governance Council:** phê duyệt và rà soát chính sách này; nhận báo cáo định kỳ về nợ kỹ thuật tồn đọng và defect escape rate ở mức tổ chức.
- **Head of Quality / QA Lead (Document Owner):** sở hữu chính sách; ban hành và cập nhật Testing Standard (Tầng 3) quy định ngưỡng coverage, ngưỡng quality gate và cơ chế cách ly flaky test cụ thể theo rủi ro.
- **Engineering Leadership:** quy định ngưỡng cảnh báo defect escape rate; hỗ trợ giải quyết xung đột giữa tốc độ phát hành và xử lý nợ kỹ thuật.
- **Product Owner / đại diện nghiệp vụ:** thống nhất cùng đội phát triển tỷ lệ dung lượng phân bổ cho nợ kỹ thuật theo Điều 5.4.2; cân đối ưu tiên giữa tính năng mới và xử lý nợ kỹ thuật.
- **Đội phát triển (bao gồm QA/kiểm thử):** áp dụng DoD, duy trì chiến lược kiểm thử theo test pyramid, ghi nhận nợ kỹ thuật, thực hiện code review chất lượng, duy trì quality risk log.
- **Release Owner / người phê duyệt phát hành:** ký xác nhận chấp nhận rủi ro chất lượng còn tồn đọng trước khi phát hành theo Điều 5.7.2.

7. Đo lường tuân thủ

COMPLIANCE MEASUREMENT

- Tỷ lệ hạng mục công việc đáp ứng đầy đủ Definition of Done tối thiểu trước khi đóng.
- Tỷ lệ pipeline có quality gate tự động đang hoạt động; số lần override quality gate và tỷ lệ có phê duyệt hợp lệ.
- Số lượng và tỷ lệ flaky test đang trong diện cách ly, thời gian trung bình để khắc phục.
- Defect escape rate theo hệ thống/đội, xu hướng theo thời gian.
- Số mục nợ kỹ thuật đang mở, tuổi trung bình (thời gian tồn đọng), và tỷ lệ xử lý theo chu kỳ phân bổ dung lượng.
- Số rủi ro chất lượng đang mở trong quality risk log, số rủi ro quá hạn kế hoạch xử lý.

8. Cơ chế ngoại lệ

EXCEPTIONS

Áp dụng theo đúng cơ chế tại Engineering Constitution, Điều 8 và SDGP, Điều 8.

9. Không tuân thủ

NON-COMPLIANCE

Áp dụng theo đúng nguyên tắc tại SDGP, Điều 9: báo cáo và khắc phục cho vi phạm thông thường; chỉ xem xét kỷ luật lao động (qua HR/Pháp chế, theo Nội quy lao động) đối với hành vi cố ý — ví dụ cố tình bỏ qua quality gate mà không xin phê duyệt, hoặc che giấu rủi ro chất lượng đã biết khỏi quality risk log tại Điều 5.7.

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

DOCUMENT OWNERSHIP & REVIEW CYCLE

Document Owner: Head of Quality / QA Lead. Chu kỳ rà soát: hằng năm, hoặc ngay khi có thay đổi đáng kể về công cụ kiểm thử, mô hình chất lượng tham chiếu, hoặc khi defect escape rate cho thấy chiến lược hiện tại không còn hiệu quả.

11. Tài liệu liên quan

RELATED DOCUMENTS

- Engineering Constitution v1.0, Software Development Governance Policy v1.0 (Tầng 1-2)
- Secure Software Development Policy, AI-Assisted Software Development Policy (Tầng 2)
- Testing Standard, Coding Standard, CI/CD Standard (Tầng 3)
- Technical Debt Register / Quality Risk Log Procedure, Production Release Procedure (Tầng 4/5)

Chất lượng không phải là điều kiểm tra được sau cùng, mà là điều được xây dựng ngay từ đầu;
một ngưỡng coverage ép cứng không thay thế được một test case biết phát hiện lỗi thật;
nợ kỹ thuật không biến mất khi bị lờ đi, chỉ tích lũy âm thầm cho đến khi phải trả giá đắt hơn;
và không rủi ro chất lượng nào được mang ra production mà chưa có ai đứng tên chấp nhận nó.

12. Phê duyệt

APPROVAL

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