QUY TRÌNH PHÁT TRIỂN PHẦN MỀM 7 GIAI ĐOẠN

Bộ đặc tả 35 hoạt động kỹ thuật vòng đời toàn trình, ma trận phân nhiệm RACI và các điểm kiểm soát cổng chất lượng (Decision Gates)

KHUNG KIẾN TRÚC QUY TRÌNH VÒNG ĐỜI QUỐC TẾ

Sơ đồ khối luồng tổng thể 7 giai đoạn (ISO/IEC/IEEE 12207:2017 & SWEBOK v4.0)

GIAI ĐOẠN 1
Xác định & Phân tích Yêu cầu Phần mềm
Bước 1 – Bước 5
Điểm then chốt: Khóa nhu cầu, phân tích tính khả thi, đặc tả SRS/Use Case, thiết lập Requirements Baseline qua Hội đồng CCB.

GIAI ĐOẠN 2
Định nghĩa Kiến trúc & Thiết kế
Bước 6 – Bước 10
Điểm then chốt: Phân rã kiến trúc ĐH/Động, thiết kế API & DB, thiết kế module chi tiết, Threat Modeling và chốt Design Baseline (PDR/CDR).

GIAI ĐOẠN 3
Xây dựng & Tích hợp Phần mềm
Bước 11 – Bước 15
Điểm then chốt: Lập trình tuân thủ an toàn, Unit Testing & Code Review, kích hoạt CI/CD và đóng và đánh giá sẵn sàng thử nghiệm (TRR).

GIAI ĐOẠN 4
Kiểm thử và xác nhận
Bước 16 – Bước 20
Điểm then chốt: Lập kế hoạch kiểm thử, xác nhận phần mềm trong giai, kiểm thử hệ thống & an ninh, nghiệm thu UAT và phê duyệt QR.

GIAI ĐOẠN 5
Chuyển giao & Triển khai
Bước 21 – Bước 25
Điểm then chốt: Đóng gói bản phát hành, chuyển đổi dữ liệu Prod, Smoke Test kiểm tra tính toàn vẹn và Ký biên bản Nghiệm thu Bàn giao (AR).

Tài liệu tham khảo & Căn cứ quy chuẩn quốc tế

CĂN CỨ QUY CHUẨN

Hệ thống quy trình phát triển phần mềm toàn trình 7 giai đoạn được xây dựng và chuẩn hóa dựa trên các tài liệu quy chuẩn kỹ thuật và cấm nang tri thức quốc tế:

- **TIÊU CHUẨN TRỌNG TÂM** — **ISO/IEC/IEEE 12207:2017 — Systems and software engineering — Software life cycle processes**
  Tiêu chuẩn quốc tế quy định khung quy trình vòng đời phần mềm toàn diện, xác định chi tiết các tiến trình kỹ thuật (Technical Processes), hoạt động thực hiện và sản phẩm bàn giao từ giai đoạn xác định nhu cầu, phân tích yêu cầu, kiến trúc, thiết kế, lập trình, kích hoạt, kiểm thử (V&V), chuyển giao, đến vận hành và bảo trì/hủy bỏ hệ thống.
- **KHUNG TRI THỨC NGÀNH** — **SWEBOK v4.0 (IEEE Computer Society) — Guide to the Software Engineering Body of Knowledge**
  Cẩm nang tri thức cốt lõi về kỹ nghệ phần mềm do IEEE Computer Society ban hành, cung cấp các ngoại lý lý thuyết, thuật ngữ chuyên ngành chuẩn hóa và các phương pháp thực hành kỹ thuật tốt nhất (best practices) cho từng chuyên môn trong dự án phần mềm.
- **TIÊU CHUẨN BỔ TRỢ** — **NIST SP 800-218 (SSDF) & ISO/IEC/IEEE 29119**
  Khung phát triển phần mềm an toàn **NIST SP 800-218 (Secure Software Development Framework)** nhằm kiểm soát an toàn chuỗi cung ứng, bảo mật mã nguồn và quản lý lỗ hổng, kết hợp bộ tiêu chuẩn quốc tế **ISO/IEC/IEEE 29119** về các quy trình, tài liệu và kỹ thuật kiểm thử phần mềm hoàn chỉnh.

GIAI ĐOẠN 1

Xác định và Phân tích Yêu cầu Phần mềm

BẢNG PHÂN NHIỆM & CÁC BƯỚC QUY CHUẨN GIAI ĐOẠN 1 — Bước 1 – Bước 5

BƯỚC	DIỄN GIẢI, MỤC TIÊU, KẾT QUẢ ĐẦU RA & RÉ NHÁNH	PHÂN CẤP TRÁCH NHIỆM (RACI)
1	**Bước 1: KHẢI THÁC YÊU CẦU (REQUIREMENTS ELICITATION)** **Mục tiêu:** Thu thập đầy đủ, chính xác nhu cầu, nguyện vọng, mục tiêu sử mệnh và các ràng buộc từ tất cả các bên liên quan (Stakeholders) cũng như từ các tài liệu quy chuẩn và hệ thống liên quan. **Cách thực thực hiện:** Nhận diện các nguồn yêu cầu (người dùng, hệ thống cũ), quy chuẩn an ninh/bảo mật, tiêu chuẩn ngành. Sử dụng các kỹ thuật như phỏng vấn sâu, bảo thảo, JRP/JAD, quan sát thực địa, khảo sát, xây dựng kịch bản Use Case / User Story và áp dụng Tư duy Thiết kế (Design Thinking) để khám phá cả các nhu cầu ẩn. **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Danh sách Nhu cầu Bên liên quan sơ bộ (Stakeholder Needs List). • Biên bản Phỏng vấn / Khảo sát & Dự thảo Kịch bản Nghiệp vụ (Use Cases / User Stories draft). **Đánh giá và xử lý:** Đánh giá tính đầy đủ của nguồn thông tin và mục đỏ bao phủ các nhóm bên liên quan. Nếu thông tin thu thập bị mâu thuẫn nặng, thiếu hụt thông tin từ các bên liên quan trọng yếu hoặc không rõ mục tiêu: Quay lại bước tái nhận diện nguồn yêu cầu và tổ chức lại các phiên làm việc JRP/JAD với Khách hàng/SMEs trước khi chuyển sang bước Phân tích.	**[Chịu trách nhiệm]:** Product Owner (PO) **[Phê duyệt]:** Business Analyst (BA) **[Tham vấn (C)]:** Project Manager (PM), Software Architect (SA), UI/UX Designer (UI/UX), Technical Lead (Tech Lead), Security Control / Tester (SC), Security Lead (Sec Lead)
2	**Bước 2: PHÂN TÍCH YÊU CẦU (REQUIREMENTS ANALYSIS)** **Mục tiêu:** Hiểu rõ ý nghĩa, mối quan hệ, tính khả thi và tác động của các yêu cầu thô; phân loại, mô hình hóa và đảm phán giải quyết các mâu thuẫn giữa các bên liên quan. **Cách thực thực hiện:** Phân loại yêu cầu thành Chức năng, Phi chức năng / Chất lượng dịch vụ (QoS) hiểu riêng, đối tin cậy), Ràng buộc Công nghệ, An toàn & Bảo mật. Xây dựng mô hình ổn gộc (UML/SysML, sơ đồ luồng dữ liệu, sơ đồ trạng thái). Phân tích tính khả thi kỹ thuật/chi phí và đảm phán giải quyết các yêu cầu xung đột. **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Mô hình Lõi-gốc Phần mềm (Software Logical Model / Sơ đồ UML/SysML). • Báo cáo Phân tích Tính khả thi & Đánh đổi (Feasibility & Trade-off Analysis Report). • Danh sách Yêu cầu đã Phân loại & Hồ sơ Giải quyết Xung đột. **Đánh giá và xử lý:** Xác nhận tính khả thi kỹ thuật và sự thống nhất của các bên liên quan về phương án đánh đổi. • Nếu yêu cầu bất khả thi về kỹ thuật hoặc vượt quá ngân sách/thời gian: Chuyển đánh phản với Khách hàng để giảm phạm vi (Descope) hoặc tìm giải pháp thay thế. • Nếu phát hiện lỗ hổng trí thức do thiếu thông tin nghiệp vụ: Quay lại Bước 1 (Khai thác Yêu cầu) để phỏng vấn bổ sung.	**[Chịu trách nhiệm]:** Product Owner (PO) **[Phê duyệt]:** Business Analyst (BA) **[Tham vấn (C)]:** Project Manager (PM), Software Architect (SA), Architect Designer (DBA), Technical Lead (Tech Lead), Quality Control / Tester (QC), Security Lead (Sec Lead)
3	**Bước 3: ĐẶC TẢ YÊU CẦU (REQUIREMENTS SPECIFICATION)** **Mục tiêu:** Chuẩn hóa toàn bộ yêu cầu đã phân tích thành tài liệu chính thức (SRS/SRS) theo tiêu chuẩn quy định, đảm bảo tăng yêu cầu có tính không mơ hồ, kiểm thử được, nguyên tắc và truy xuất được. **Cách thực thực hiện:** Biên soạn tài liệu bằng Ngôn ngữ tự nhiên có cấu trúc, mô hình hóa hoặc kịch bản kiểm thử nghiệm thu (BDD/ATDD / Acceptance Criteria). Gán các thuộc tính phụ trợ cho từng yêu cầu (Mã định danh ID duy nhất, Mức độ ưu tiên, Lý do tồn tại/Rationale, Tiêu chí kiểm thử nghiệm thủ) và lập Ma trận Truy xuất (Traceability Matrix). **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Dự thảo Đặc tả Yêu cầu Phần mềm (Draft SRS - Software Requirements Specification). • Dự thảo Tài liệu Yêu cầu Giao diện (Draft IRD / ICD). • Ma trận Truy xuất Yêu cầu sơ bộ (Preliminary Requirements Traceability Matrix). **Đánh giá và xử lý:** Tránh giả định hoàn thiện và chất lượng văn bản đặc tả (kiểm tra các thuộc tính Unambiguous, Testable, Atomic, Complete). • Nếu văn bản còn mơ hồ, thiếu tiêu chí kiểm thử hoặc không thể truy xuất: Yêu cầu BA chỉnh sửa, chuẩn hóa lại văn bản. • Nếu phát hiện mâu thuẫn logic mới phát sinh: Quay lại Bước 2 (Phân tích Yêu cầu) để tái phân tích.	**[Chịu trách nhiệm]:** Product Owner (PO) **[Phê duyệt]:** Business Analyst (BA) **[Tham vấn (C)]:** Project Manager (PM), Software Architect (SA), Architect Designer (DBA), Technical Lead (Tech Lead), Software Quality Assurance (SQA), Quality Control / Tester (QC), Software Configuration Manager (SCM), Security Lead (Sec Lead)
4	**Bước 4: THẨM ĐỊNH YÊU CẦU (REQUIREMENTS VALIDATION)** **Mục tiêu:** Xác nhận và đạt được sự đồng thuận chính thức từ Khách hàng/Bên liên quan rằng bản đặc tả SRS thể hiện chính xác và đầy đủ nhu cầu thực tế trước khi phê duyệt. **Cách thực thực hiện:** Tổ chức đạt rà soát chính thức (Software Requirements Review - SWRR / Formal Inspection) đa góc nhìn (Khách hàng, Lập trình viên, Kiểm thử viên). Sử dụng các bản mẫu UI/UX (Prototyping) hoặc mô phỏng (Simulation) để khách hàng trực tiếp kiểm tra và xác nhận. **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Biên bản Đánh giá Yêu cầu Phần mềm (SWRR Review Minutes / Inspection Report). • Danh mục Yêu cầu Điều chỉnh / Phiếu Nhận xét RIDs (Review Item Discrepancies). • Bản Đặc tả Yêu cầu Phần mềm đã qua thẩm định (Validated SRS). **Đánh giá và xử lý:** Quyết định phê duyệt bản Đặc tả Yêu cầu Phần mềm (Đạt / Đạt có điều kiện / Không đạt). • Nếu sai lệch do hiểu lầm nghiệp vụ cần bổ hoặc thiếu tính năng quan trọng: Quay lại Bước 1 & Bước 2 để khai thác lại và tái phân tích. • Nếu chỉ là lỗi diễn đạt hoặc mô tả chưa rõ ràng: Quay lại Bước 3 (Đặc tả) để cập nhật tại văn bản trước khi rà soát lại.	**[Chịu trách nhiệm]:** Product Owner (PO) **[Phê duyệt]:** Business Analyst (BA), Quality Control / Tester (QC) **[Tham vấn (C)]:** Project Manager (PM), Software Architect (SA), UI/UX Designer (UI/UX), Architect Designer (DBA), Technical Lead (Tech Lead), Software Quality Assurance (SQA), Security Lead (Sec Lead)
5	**Bước 5: QUẢN LÝ YÊU CẦU & THIẾT LẬP CƠ SỞ (REQUIREMENTS MANAGEMENT & BASELINING)** **Mục tiêu:** Phê duyệt chính thức Tập yêu cầu cơ sở (Requirements Baseline), viện hành quy trình kiểm soát Yêu cầu Thay đổi (CR) và đưa tri ma trận truy xuất xuyên suốt vòng đời phát triển. **Cách thực thực hiện:** Trình bộ hồ sơ SRS/IRD đã thẩm định lên Hội đồng Kiểm soát Cấu hình (CCB) để chốt Baseline. Vận hành quy trình kiểm soát thay đổi (tiếp nhận CR, phân tích tác động về chi phí/lịch trình/kỹ thuật, CCB quyết định, cập nhật baseline và thông báo cho các bên). **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Tập yêu cầu Cơ sở đã phê duyệt (Approved Requirements Baseline - RB / Baselined SRS). • Ma trận Truy xuất Yêu cầu Hoàn chỉnh (Full Traceability Matrix). • Nhật ký Thay đổi Yêu cầu & Biên bản Quyết định của CCB (CR Log / CCB Decisions). **Đánh giá và xử lý:** CCB quyết định phê duyệt đóng gói Baseline hoặc Phê duyệt / Bác bỏ mỗi Yêu cầu Thay đổi (CR). Xử lý khi có Yêu cầu Thay đổi (ECR) phát sinh sau Baseline: • Nếu CCB Từ chối (Reject): Giữ nguyên Baseline hiện tại, phản hồi lý do cho bên đề xuất. • Nếu CCB Chấp thuận (Approve): Kích hoạt luồng cập nhật bằng cách Quay lại Bước 2 (Phân tích Tác động) → Bước 3 (Đặc tả lại) → Bước 4 (Thẩm định lại phạm vi thay đổi) → Cập nhật và phát hành phiên bản Baseline mới (v1.1, v2.0...) — HẾT —	**[Chịu trách nhiệm]:** Project Manager (PM) **[Phê duyệt]:** Business Analyst (BA), Software Configuration Manager (SCM) **[Tham vấn (C)]:** Product Owner (PO), Software Quality Assurance (SQA)

SƠ ĐỒ LUỒNG KỸ THUẬT GIAI ĐOẠN 1

GIAI ĐOẠN 2

Định nghĩa Kiến trúc và Thiết kế Phần mềm

BẢNG PHÂN NHIỆM & CÁC BƯỚC QUY CHUẨN GIAI ĐOẠN 2 — Bước 6 – Bước 10

BƯỚC	DIỄN GIẢI, MỤC TIÊU, KẾT QUẢ ĐẦU RA & RÉ NHÁNH	PHÂN CẤP TRÁCH NHIỆM (RACI)
6	**Bước 6: PHÂN RÃ & ĐỊNH NGHĨA KIẾN TRÚC PHẦN MỀM (SOFTWARE ARCHITECTURAL SYNTHESIS & DECOMPOSITION)** **Mục tiêu:** Chuyển đổi các yêu cầu chức năng, phi chức năng và ràng buộc từ Đặc tả (SRS) thành cấu trúc kiến trúc tổng thể (tĩnh và động), lựa chọn kiểu kiến trúc và phân rã hệ thống thành các mô-đun/thành phần cấp cao. **Cách thực thực hiện:** Lựa chọn kiến trúc phù hợp (Architectural Styles/Patterns: Layered, Microservices, Event-Driven, Component-Based...). Phân rã cấu trúc tĩnh (Packages, Subsystems, Classes) và cấu trúc động (Tasks, Threads, Processes, Sequence Diagrams). Đánh giá sự đánh đổi giữa các chỉ số chất lượng (Hiệu năng vs. Bảo mật vs. Tính dễ bảo trì). **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Dự thảo Tài liệu Mô tả Kiến trúc Phần mềm (Software Architecture Document - SAD). • Bộ Sơ đồ Kiến trúc Tĩnh & Động (UML Component Diagram, Deployment Diagram, Sequence Diagram). **Đánh giá và xử lý:** Đánh giá tính đầy đủ của kiến trúc đối với các chỉ số chất lượng (QoS) và tính khả thi về nhật ký hàng công nghệ. Nếu kiến trúc không đáp ứng được chỉ số hiệu năng hoặc độ rủi ro bảo mật: Tiến hành thiết kế lại phương án kiến trúc mới (Nhà máng lại architectural patterns). Nếu phát hiện yêu cầu SRS chưa rõ ràng/mâu thuẫn khi thử ở cấp kiến trúc: Quay lại Giai đoạn 1 (Đặc tả Tính / Bước 3: Đặc tả lại SRS) để điều chỉnh phạm vi.	**[Chịu trách nhiệm]:** Project Manager (PM) **[Phê duyệt]:** Software Architect (SA) **[Tham vấn (C)]:** UI/UX Designer (UI/UX), Database Administrator (DBA), Technical Lead (Tech Lead), Software Quality Assurance (SQA), DevOps Engineer (DevOps), Site Reliability Engineer (SRE), System Administrator (SysAdmin), Security Lead (Sec Lead)
7	**Bước 7: THIẾT KẾ GIAO DIỆN VÀ MÔ HÌNH DỮ LIỆU (INTERFACE & DATA DESIGN)** **Mục tiêu:** Định nghĩa chuẩn hóa toàn bộ các giao diện trong tác bên ngoài (External APIs/ICDs) và giao diện nội bộ giữa các thành phần phần mềm, cùng với thiết kế cấu trúc dữ liệu loàn cục/CSDL. **Cách thực thực hiện:** Thiết kế giao diện ngoại vi (với phần cứng, hệ thống bên thứ ba, giao diện mạng) và giao diện nội bộ (bản tin, API endpoints, gRPC/REST protocols). Thiết kế mô hình dữ liệu (Conceptual/Logical/Physical Data Model), cấu trúc CSDL (Relational / NoSQL), chiến lược lưu trữ và bảo mật dữ liệu. **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Tài liệu Kiến soát Giao diện (Interface Control Document - ICD / API Specifications). • Tài liệu Thiết kế Cơ sở Dữ liệu & Sơ đồ Mô hình Dữ liệu (Database Design Document & ERD/Data Models). **Đánh giá và xử lý:** Đánh giá độ tương thích giao diện, tính toàn vẹn dữ liệu và chuẩn an ninh truyền thông (TLS, OAuth2...). Nếu giao diện xung đột với hệ thống bên ngoài hoặc không đảm bảo bảng thông-dữ liệu: Yêu cầu thiết kế lại giao thức và định dạng dữ liệu (Payload format/protocol revision). Nếu cấu trúc CSDL không đảm bảo tính nhất quán (ACID) hoặc khả năng mở rộng: Yêu cầu tinh chỉnh lại sơ đồ ERD hoặc chiến lược phân tán CSDL.	**[Chịu trách nhiệm]:** Product Owner (PO) **[Phê duyệt]:** UI/UX Designer (UI/UX) **[Tham vấn (C)]:** Project Manager (PM), Software Architect (SA), Technical Lead (Tech Lead), Front-End Developer (FE Dev), Back-End Developer (BE Dev), Full-Stack Developer (FS Dev), Mobile Developer (Mobile Dev), Quality Control / Tester (QC)
8	**Bước 8: THIẾT KẾ CHI TIẾT THÀNH PHẦN VÀ THUẬT TOÁN (DETAILED COMPONENT DESIGN)** **Mục tiêu:** Phân rã các thành phần kiến trúc cấp cao thành các đơn vị phần mềm (Software Units) chi tiết có thể lập trình được, bao gồm cấu trúc thuật toán, xử lý ngoại lệ và quản lý tài nguyên. **Cách thực thực hiện:** Chi tiết hóa thuật toán nội bộ của từng hàm/lớp/mô-đun (sử dụng Pseudo-code, Activity Diagrams, Statechart Diagrams). Thiết kế cơ chế xử lý lỗi/ngoại lệ (Exception handling), quản lý bộ nhớ, tiến trình song song (Concurrency) và thiết lập tiêu chuẩn mã hóa chi tiết (Coding Standards, SOLID). **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Tài liệu Mô tả Thiết kế Chi tiết Phần mềm (Software Design Description - SDD hoàn chỉnh). • Sơ đồ Chi tiết Thuật toán & Trạng thái (Detailed Activity & Statechart Diagrams). **Đánh giá và xử lý:** Đánh giá tính khả thi trong tập trình, mức độ hoàn thiện thuật toán và tính kiểm thử được của các đơn vị (Unit Testability). Nếu thuật toán quá phức tạp, rủi ro cao nghiêm trọng hiệu năng hoặc có rò rỉ tài nguyên: Yêu cầu tối ưu hóa cấu trúc dữ liệu / thuật toán. Nếu phát hiện ở mức nguyên tắc thiết kế SOLID hoặc coupling quá cao: Refactor lại thiết kế thành phần trước khi cho phép bắt đầu code.	**[Chịu trách nhiệm]:** Software Architect (SA) **[Phê duyệt]:** Database Architect (DBA) **[Tham vấn (C)]:** Technical Lead (Tech Lead), Back-End Developer (BE Dev), DevOps Engineer (DevOps), Site Reliability Engineer (SRE), Security Lead (Sec Lead)
9	**Bước 9: RÀ SOÁT THIẾT KẾ, PHÂN TÍCH AN NINH VÀ MÔ HÌNH HÓA MỐI ĐE DỌA (DESIGN REVIEW & THREAT MODELING)** **Mục tiêu:** Thẩm định độc lập toàn bộ bản thiết kế kiến trúc và thiết kế chi tiết nhằm phát hiện sớm các lỗi thiết kế, ngăn cơ an ninh thông tin và đánh giá cái rủi ro kỹ thuật trước khi viết code. **Cách thực thực hiện:** Thực hiện mô hình hóa mối đe dọa an ninh (Threat Modeling - ví dụ Mô hình STRIDE/PASTA) để tìm lỗ hổng kiến trúc. Tổ chức các đợt rà soát thiết kế chính thức (Software Design Review / Architecture Review) với các chuyên gia độc lập, phân tích rủi ro kỹ thuật và đánh giá khả năng duy trì/bảo trì dự án mã nguồn. **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Báo cáo Mô hình hóa Mối đe dọa An ninh (Threat Modeling Report). • Biên bản Rà soát Thiết kế Phần mềm (Design Review Minutes / Architecture Review Report). • Danh mục Phiếu Yêu cầu Điều chỉnh Thiết kế (Design Discrepancies / RIDs Log). **Đánh giá và xử lý:** Đánh giá phê duyệt hồ sơ Kiến trúc và Thiết kế chi tiết (Conditional Pass / Fail). Nếu phát hiện lỗ hổng an ninh hiện trạc nghiêm trọng hoặc vi phạm tăng bước kỹ thuật cốt lõi: Hội đồng Kiến trúc bác bỏ phê duyệt: Quay lại Bước 1 hoặc Bước 3 của Giai đoạn này để thiết kế lại trước khi tới chốt cả soát lại.	**[Chịu trách nhiệm]:** Software Architect (SA) **[Phê duyệt]:** Technical Lead (Tech Lead), Back-End Developer (BE Dev) **[Tham vấn (C)]:** Database Administrator (DBA), Front-End Developer (FE Dev), Mobile Developer (Mobile Dev), Quality Control / Tester (QC), Security & Performance Tester (Sec/Perf Tester), Security Lead (Sec Lead)
10	**Bước 10: LẬP KẾ HOẠCH TÍCH HỢP VÀ VÁ CHỐT CƠ SỞ THIẾT KẾ (DESIGN BASELINING & STRATEGY)** **Mục tiêu:** Xây dựng chiến lược tích hợp phần mềm, kế hoạch xác minh & xác nhận (V&V), và phê duyệt chính thức Tập cơ sở thiết kế (Design Baseline - PDR/CDR) làm chuẩn cho Giai đoạn Xây dựng. **Cách thực thực hiện:** Xây dựng Chiến lược Tích hợp Phần mềm (Top-down, Bottom-up, Continuous Integration). Soạn thảo Kế hoạch Xác minh & Xác nhận (SVaR: SVaVP). Tính toán bộ Hồ sơ Thiết kế (SAD, SDD, ICD, Threat Model) lên Hội đồng Kiểm soát Cấu hình (CCB) hoặc Ban Đánh giá Thiết kế Sơ bộ/Chi tiết (PDR/CDR) để phê duyệt chốt Baseline. **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Kế hoạch Tích hợp Phần mềm (Software Integration Plan - SIP/TP). • Kế hoạch Xác minh & Xác nhận Phần mềm (Software Verification & Validation Plan - SVaP/SVaVP). • Tập cơ sở Thiết kế được phê duyệt (Approved Design Baseline / Baselined SAD & SDD). **Đánh giá và xử lý:** Hội đồng Thẩm định cấp phép đóng gói Baseline Thiết kế tại cột mốc PDR/CDR. Nếu không đạt cột mốc PDR/CDR: Không cấp phép chuyển sang Giai đoạn 3 (Xây dựng). Quay lại các bước thiết kế trước đó để hoàn thiện hồ sơ. Khi có yêu cầu thay đổi thiết kế sau Baseline: Kích hoạt quy trình CCB để kiểm soát cập nhật phiên bản kiến trúc.	**[Chịu trách nhiệm]:** Project Manager (PM) **[Phê duyệt]:** Software Architect (SA), Software Configuration Manager (SCM) **[Tham vấn (C)]:** UI/UX Designer (UI/UX), Database Administrator (DBA), Technical Lead (Tech Lead), Quality Control / Tester (QC), Security Lead (Sec Lead), DevOps Engineer (DevOps), Site Reliability Engineer (SRE), System Administrator (SysAdmin), Security Lead (Sec Lead)

SƠ ĐỒ LUỒNG KỸ THUẬT GIAI ĐOẠN 2

GIAI ĐOẠN 3

Xây dựng / Lập trình và Tích hợp Phần mềm

BẢNG PHÂN NHIỆM & CÁC BƯỚC QUY CHUẨN GIAI ĐOẠN 3 — Bước 11 – Bước 15

BƯỚC	DIỄN GIẢI, MỤC TIÊU, KẾT QUẢ ĐẦU RA & RÉ NHÁNH	PHÂN CẤP TRÁCH NHIỆM (RACI)
11	**Bước 11: LẬP TRÌNH VÀ TUÂN THỦ TIÊU CHUẨN MÃ NGUỒN AN TOÀN (SECURE & STANDARDIZED CODING)** **Mục tiêu:** Hiện thực hóa bản thiết kế chi tiết (SDD) thành các đơn vị mã nguồn (Software Units) chất lượng cao, tuân thủ nghiêm ngặt quy chuẩn mã hóa và tiêu chuẩn an toàn thông tin. **Cách thực thực hiện:** Lập trình mã nguồn theo các tiêu chuẩn mã hóa quy định (Coding Standards), áp dụng các thực hành lập trình an toàn (Secure Coding Practices - kiểm soát đầu vào, mã hóa dữ liệu, phòng chống OWASP Top 10), và tắt sử dụng các thư viện mã nguồn mở/bên thứ ba đã qua kiểm duyệt an toàn (SBOM). **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Mã nguồn phần mềm (Source Code) của từng đơn vị phần mềm. • Tập tin Cấu hình & Danh mục Thành phần Tài sản dùng (Configuration Files & SBOM - Software Bill of Materials). **Đánh giá và xử lý:** Xác nhận mã nguồn tuân thủ Coding Standards và Quy trình mã nguồn an toàn. Nếu mã nguồn vi phạm quy chuẩn mã hóa hoặc sử dụng các thư viện cấm/rủi lỗng bảo mật đã biết (CVE): Từ chối kích hoạt, yêu cầu lập trình viên hiệu chỉnh và quét lại mã nguồn cho đến khi đạt chuẩn.	**[Chịu trách nhiệm]:** Software Architect (SA) **[Phê duyệt]:** Technical Lead (Tech Lead), DevOps Engineer (DevOps) **[Tham vấn (C)]:** Database Administrator (DBA), Front-End Developer (FE Dev), Back-End Developer (BE Dev), Full-Stack Developer (FS Dev), Mobile Developer (Mobile Dev), Software Quality Assurance (SQA), Quality Control / Tester (QC), Software Configuration Manager (SCM), Security Lead (Sec Lead)
12	**Bước 12: KIỂM THỬ ĐƠN VỊ VÀ PHÂN TÍCH MÃ NGUỒN (UNIT TESTING & CODE ANALYSIS / REVIEW)** **Mục tiêu:** Xác minh tính đúng đắn logic của từng đơn vị phần mềm độc lập, phát hiện sớm lỗi cú pháp, rò rỉ bộ nhớ và các lỗ hổng an ninh mã nguồn trước khi thực hiện tích hợp. **Cách thực thực hiện:** Xây dựng và chạy các kịch bản kiểm thử đơn vị tự động (Automated Unit Tests) đạt tỷ lệ bao phủ mã (Code Coverage: Statement/Branch Coverage). Thực hiện rà soát mã đồng nghiệp (Peer Review) và ở quét mã tự động bằng công cụ phân tích tĩnh (SAST - Static Application Security Testing). **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Bộ Kịch bản Kiểm thử Đơn vị (Unit Test Cases / Scripts). • Báo cáo Rà soát Kiểm thử Đơn vị & Tỷ lệ Bao phủ Mã (Unit Test Report & Code Coverage Report). • Báo cáo Phân tích Tĩnh Mã Nguồn & Biên bản Rà soát Mã (SAST Scan Report & Peer Review Minutes). **Đánh giá và xử lý:** Tỷ lệ bao phủ kiểm thử đơn vị >= 80% (hoặc ngưỡng quy định) và không còn lỗi Severity High/Critical. • Nếu tỷ lệ bao phủ chưa đạt hoặc có unit test thất bại: Trả lại mã nguồn cho lập trình viên bổ sung test case và sửa lỗi. • Nếu phát hiện lỗi logic thiết kế cấp thấp: Quay lại Giai đoạn 2 (Bước 3: Thiết kế Chi tiết) để hiệu chỉnh thuật toán trước khi viết tại mã nguồn.	**[Chịu trách nhiệm]:** Technical Lead (Tech Lead) **[Phê duyệt]:** Front-End Developer (FE Dev), Back-End Developer (BE Dev), Full-Stack Developer (FS Dev), Mobile Developer (Mobile Dev) **[Tham vấn (C)]:** Scrum Master (SM), Software Architect (SA), UI/UX Designer (UI/UX), Database Administrator (DBA), Software Configuration Manager (SCM)
13	**Bước 13: TÍCH HỢP PHẦN MỀM VÀ TỰ ĐỘNG HÓA BUILD (SOFTWARE INTEGRATION & CONTINUOUS BUILD)** **Mục tiêu:** Hợp nhất các đơn vị phần mềm đơn lẻ thành các cụm thành phần và bảo đảng gói hệ thống hoàn chỉnh theo chiến lược tích hợp đã lập kế hoạch. **Cách thực thực hiện:** Áp dụng quy trình Tích hợp Liên tục (CI - Continuous Integration). Thực hiện hợp nhất mã nguồn (Merge Code), xử lý xung đột (Merge Conflicts), biên dịch tự động (Automated Build) và tạo bản đóng gói thực thi (Executable Packages / Binary Artifacts / Docker Images). **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Bản đóng gói Phần mềm Thực thi (Executable Build Packages / Binary Artifacts / Docker Images). • Nhật ký Biên dịch & Tích hợp (Build & Integration Logs). • Bản ghi Nhận Cấu hình Bản dựng (Configuration Build Baseline / Tagged Release Candidate). **Đánh giá và xử lý:** Bản build tự động thành công và các nhận tính toán vẹn của tập thực thi. Nếu biên dịch thất bại (Build Broken) hoặc xung đột phát hiện mã nguồn: Lập tức chặn tiến trình CI, thông báo cho lập trình viên vừa commit để sửa lỗi khẩn cấp trong vòng 30 phút hoặc revert commit vệ phiên bản đó ngăn gián nhiễu.	**[Chịu trách nhiệm]:** Technical Lead (Tech Lead) **[Phê duyệt]:** Front-End Developer (FE Dev), Back-End Developer (BE Dev), Full-Stack Developer (FS Dev), Mobile Developer (Mobile Dev) **[Tham vấn (C)]:** Software Quality Assurance (SQA), Quality Control / Tester (QC), Automation Test Engineer (Auto Test), DevOps Engineer (DevOps)
14	**Bước 14: KIỂM THỬ TÍCH HỢP PHẦN MỀM (SOFTWARE INTEGRATION TESTING)** **Mục tiêu:** Xác minh khả năng tương tác, giao tiếp và truyền nhận dữ liệu chính xác giữa các đơn vị/thành phần phần mềm sau khi tích hợp. **Cách thực thực hiện:** Lập kịch bản và thực thi kiểm thử giao diện nội bộ (Internal API Testing, Component Integration Testing). Thử nghiệm kiểm tra luồng dữ liệu liên mô-đun, quản lý tài nguyên đồng thời và các kịch bản ngoại lệ khi các mô-đun phối hợp hoạt động. **SẢN PHẨM BÀN GIAO QUY CHUẨN:** • Kịch bản & Dữ liệu Kiểm thử Tích hợp (Integration Test Cases & Data Sets). • Báo cáo Kết quả Kiểm thử Tích hợp Phần mềm (Software Integration Test Report). • Nhật ký Lỗi Tích hợp (Integration Defect Log / PRs). **Đánh giá và xử lý:** 100% các ca kiểm thử tích hợp chính vượt qua thành công, không có lỗi gián đoạn hệ thống nghiêm trọng. • Nếu phát hiện lỗi giao tiếp giữa các mô-đun: Phân loại lỗi thuộc về mô-đun nào để yêu cầu lập trình viên tương ứng sửa chữa (Quay lại Bước 13). • Nếu lỗi do hiểu sai đặc tả giao diện (API Spec mismatch): Quay lại Giai đoạn 2 (Bước 2: Thiết kế Giao diện & CSDL) để đồng bộ hóa lại định nghĩa API.	**[Chịu trách nhiệm]:** Software Architect (SA) **[Phê duyệt]:** Technical Lead (Tech Lead), Security Lead (Sec Lead) **[Tham vấn (C)]:** Front-End Developer (FE Dev