



AWS First Cloud AI Journey – Project Plan

[Mambo] – [FPT University] – [Mini-Market]
[8/9/2025-12/12/2025]

TABLE OF CONTENTS

| | | |
|----------|---|----------|
| 1 | BACKGROUND AND MOTIVATION..... | 3 |
| 1.1 | EXECUTIVE SUMMARY..... | 3 |
| 1.2 | PROJECT SUCCESS CRITERIA..... | 3 |
| 1.3 | ASSUMPTIONS..... | 3 |
| 2 | SOLUTION ARCHITECTURE / ARCHITECTURAL DIAGRAM..... | 3 |
| 2.1 | TECHNICAL ARCHITECTURE DIAGRAM..... | 3 |
| 2.2 | TECHNICAL PLAN..... | 4 |
| 2.3 | PROJECT PLAN..... | 4 |
| 2.4 | SECURITY CONSIDERATIONS..... | 4 |
| 3 | ACTIVITIES AND DELIVERABLES..... | 5 |
| 3.1 | ACTIVITIES AND DELIVERABLES..... | 5 |
| 3.2 | OUT OF SCOPE..... | 5 |
| 3.3 | PATH TO PRODUCTION..... | 5 |
| 4 | EXPECTED AWS COST BREAKDOWN BY SERVICES..... | 5 |
| 5 | TEAM..... | 6 |
| 6 | RESOURCES & COST ESTIMATES..... | 6 |
| 7 | ACCEPTANCE..... | 7 |

1 BACKGROUND AND MOTIVATION

1.1 EXECUTIVE SUMMARY

Mini-Market là một dự án học thuật mô phỏng bài toán chuyển đổi số cho mô hình mini-market truyền thống tại Việt Nam. Khách hàng giả định là một chuỗi mini-market nhỏ, hiện đang vận hành chủ yếu bằng quy trình thủ công (sổ sách, Excel, bán hàng offline).

Business objectives:

- Giảm thất thoát hàng hóa và sai sót trong quản lý kho (do kiểm kê thủ công).
- Mở thêm kênh doanh thu online thông qua nền tảng e-commerce.
- Rút ngắn thời gian thanh toán, cải thiện trải nghiệm khách hàng tại quầy.
- Tạo nền tảng để mở rộng thêm nhiều chi nhánh mini-market sau này.

Technical objectives:

- Xây dựng kiến trúc .NET 3-layers (3-tier) hiện đại, dễ bảo trì, áp dụng **Repository Pattern & Unit of Work Pattern**.
- Ứng dụng các dịch vụ **AWS Managed Services** (Elastic Beanstalk, RDS, S3, CloudFront, WAF, ElastiCache) theo **AWS Well-Architected Framework**.
- Tự động hóa **CI/CD** với AWS CodePipeline & CodeBuild tích hợp với GitHub.
- Tối ưu chi phí bằng cách tận dụng **AWS Free Tier** trong năm đầu tiên.

Web e-commerce mini-market với các chức năng chính: quản lý sản phẩm, danh mục, gio hàng, đơn hàng, user account, và quản lý kho cơ bản.

- Thiết kế **solution architecture** trên nền tảng AWS.
- Triển khai các dịch vụ AWS cần thiết cho môi trường Staging và Production.
- Thiết kế & phát triển ứng dụng **ASP.NET Core MVC** theo mô hình 3-lớp.
- Thiết lập CI/CD pipeline với AWS CodePipeline / CodeBuild.
- Cấu hình các dịch vụ bảo mật cơ bản: VPC, Security Group, WAF, Private Subnet, NAT Gateway.

1.2 PROJECT SUCCESS CRITERIA

Dự án được xem là thành công nếu các tiêu chí định lượng sau được đáp ứng (ở môi trường demo / POC):

Business success criteria

- Giảm ≥ 90% sai lệch trong quản lý kho so với mô hình số tay / Excel (được đo bằng số lượng chênh lệch giữa tồn kho hệ thống và tồn kho thực tế trong các lần kiểm tra thử).
- Giảm ≥ 50% thời gian thanh toán tại quầy (từ lúc bắt đầu scan sản phẩm đến lúc in hóa đơn) so với quy trình thủ công giả lập.
- Đạt ≥ 20% doanh thu giả lập đến từ kênh online (so với tổng doanh thu mô phỏng) trong vòng 6 tháng nếu triển khai thực tế.

Technical success criteria

- Uptime (trên môi trường demo) ≥ 99.9% trong thời gian kiểm thử.
- Thời gian tải trang chủ (product listing) trung bình < 2 giây khi sử dụng CloudFront và ElastiCache (test với kịch bản tải mô phỏng).
- CI/CD pipeline triển khai thành công ≥ 90% số lần push code lên nhánh main mà không cần thao tác tay.
- Mọi secret/connection string không được hard-code trong source code mà được quản lý qua **environment variables** của Elastic Beanstalk.

1.3 ASSUMPTIONS

Các giả định chính để thực thi dự án:

- **AWS Account:** Khách hàng (hoặc nhóm dự án) cung cấp một AWS Account có quyền tạo các dịch vụ: VPC, EC2/Beanstalk, RDS, S3, CloudFront, WAF, ElastiCache, CodePipeline, CodeBuild.
- **Free Tier:** Hạ tầng sử dụng **AWS Free Tier** cho năm đầu, bao gồm EC2 t3.micro/t3a.small, RDS db.t3.micro (SQL Server Express), ElastiCache t4g.micro, S3 và CloudFront.
- **Traffic:** Lưu lượng truy cập giai đoạn đầu thấp đến trung bình (môi trường demo, chưa phải hệ thống production với hàng chục nghìn user).
- **Data volume:** Số lượng sản phẩm và đơn hàng ở mức vừa phải (vài nghìn SKUs, vài nghìn đơn hàng/tháng) – phù hợp với kiến trúc đã đề xuất.
- **Team availability:** Nhóm có đủ thời gian làm việc trong khung thời gian [08/09/2025 – 12/12/2025].
- **External dependencies:**
 - Không có hệ thống mainframe hoặc hệ thống on-premises legacy cần tích hợp.
 - Tích hợp với cổng thanh toán thực tế (VNPay, MoMo, v.v.) **không nằm trong scope chính** – chỉ mô phỏng thanh toán (ví dụ: COD).
- **Risks & constraints:**
 - Hạn chế về thời gian (dự án học kỳ).
 - Một số dịch vụ AWS có thể thay đổi giá hoặc chính sách Free Tier trong tương lai.

2 SOLUTION ARCHITECTURE / ARCHITECTURAL DIAGRAM

2.1 TECHNICAL ARCHITECTURE DIAGRAM

High-level architecture description

Proposed solution sử dụng kiến trúc 3-tier (.NET) kết hợp với hạ tầng AWS theo Well-Architected Framework:

- **Presentation Layer (WebShop):**
 - ASP.NET Core MVC 9.0
 - Chạy trên AWS Elastic Beanstalk (Windows/Linux platform cho .NET)
 - Giao tiếp với Application Layer thông qua services nội bộ trong cùng project/solution.
- **Application Layer (Services):**
 - Chịu business logic: ProductService, OrderService, UserService, InventoryService,...
 - Áp dụng **Repository Pattern & Unit of Work Pattern** để tách biệt business logic và data access.
- **Persistence Layer (Data Access):**
 - Sử dụng **Entity Framework Core** kết nối tới Amazon RDS for SQL Server.
 - Repositories cho từng aggregate: ProductRepository, OrderRepository,...

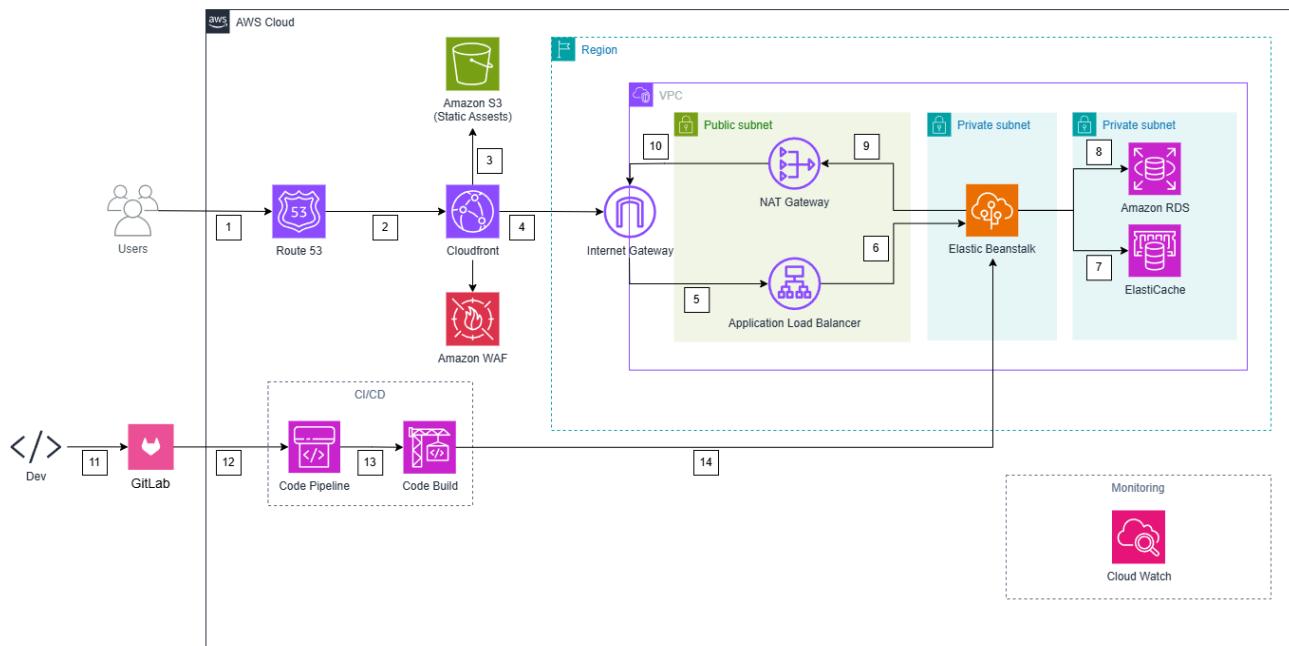
AWS Infrastructure components

- **Amazon Route 53:** Quản lý DNS và domain của ứng dụng mini-market.
- **Amazon CloudFront:** CDN cache static content (hình ảnh sản phẩm, JS, CSS) từ S3 để tăng tốc tải trang.
- **Amazon S3:** Lưu trữ static assets (product images, css, js).
- **Amazon Elastic Beanstalk:**
 - Host ứng dụng ASP.NET Core MVC.
 - Tự động tạo EC2, Auto Scaling Group, Application Load Balancer.
 - Đặt trong **Private Subnet** để tăng bảo mật, chỉ truy cập thông qua ALB.
- **Amazon RDS for SQL Server (Express):**
 - DB cho toàn bộ hệ thống.
 - Đặt trong Private Subnet, không public internet.
 - Tự động backup, patch, hỗ trợ Multi-AZ (có thể bật nếu cần).
- **Amazon ElastiCache (Redis):**
 - Cache dữ liệu nóng: danh sách sản phẩm, thông tin category, cấu hình chung.
 - Giảm tải truy vấn lặp lại lên RDS.
- **AWS WAF:**

- Gắn với CloudFront để chặn các tấn công phổ biến (SQL injection, XSS, bad bots).
- AWS VPC + Subnets + NAT Gateway + Internet Gateway:
 - VPC riêng cho hệ thống.
 - Public Subnet: ALB, NAT Gateway, CloudFront edge (bên ngoài).
 - Private Subnet: Elastic Beanstalk instances, RDS, ElastiCache.
 - NAT Gateway dùng cho instances trong Private Subnet ra Internet (update, patches).
- AWS CodePipeline + AWS CodeBuild:
 - CI/CD từ GitHub → Build → Deploy lên Elastic Beanstalk.

Tools proposed

- Visual Studio
- Git & GitHub
- AWS Management Console
- (Tùy chọn) AWS CloudFormation hoặc Elastic Beanstalk configuration templates
- Docker Desktop (cho local test)



2.2 TECHNICAL PLAN

Nhóm sẽ:

- Thiết lập .NET solution 3-layer gồm các projects: Domain, Application, Persistence, WebShop.
- Xây dựng code tuân thủ Repository + Unit of Work để đảm bảo có thể test và maintain dễ dàng.
- Sử dụng GitHub làm source control, với các nhánh: main, develop và feature branches.
- Thiết lập CI/CD Pipeline bằng AWS CodePipeline + CodeBuild:
 - Trigger: Push lên nhánh main trên GitHub.
 - CodeBuild: Build solution .NET, chạy unit tests.
 - Artifacts: .zip package cho Elastic Beanstalk.
 - Deploy: CodePipeline deploy tự động lên môi trường Staging và (sau khi approve) lên Production.

Một số cấu hình sẽ được thực hiện thủ công qua AWS Console (phù hợp với phạm vi học kỳ), bao gồm:

- Tạo VPC, Subnets, Security Groups.
- Tạo RDS, ElastiCache, S3, WAF, CloudFront.
- Kết nối domain từ Route 53 vào CloudFront/ALB.

Mọi đường đi quan trọng (critical paths) như: Đăng ký, Đăng nhập, Thêm vào giỏ, Thanh toán, Tạo đơn hàng sẽ được viết test-cases và kiểm thử kỹ trên môi trường Staging.

2.3 PROJECT PLAN

Nhóm sẽ áp dụng mô hình **Agile Scrum-like** với các sprint ngắn (1–2 tuần) trong tổng thời gian **12 tuần**, chia thành 4 Phase:

- **Phase 1 – Assessment & Foundation (Week 1–4)**
 - Phân tích yêu cầu, chốt use case.
 - Thiết kế kiến trúc logic & physical.
 - Thiết lập .NET solution, GitHub repo, AWS VPC + Subnets.
- **Phase 2 – Core Feature Development (Week 5–8)**
 - Phát triển các tính năng chính (Product, Cart, Order, Auth).
 - Xây dựng Application Services, Repositories, Unit of Work.
 - Viết Unit Tests cho business logic.
- **Phase 3 – Cloud Preparation & Refactoring (Week 9–11)**
 - Tái cấu trúc mã nguồn để tương thích Cloud.
 - Chuyển đổi hardcoded sang Environment Variables.
 - Viết buildspec.yml và chuẩn bị Dockerfile.

- **Phase 4 – Infrastructure Deployment & Go-live (Week 12)**

- Khởi tạo toàn bộ tài nguyên AWS (Beanstalk, RDS, ElastiCache, S3).
- Cấu hình Security (WAF, CloudFront)..
- Kích hoạt CI/CD Pipeline.
- UAT & Monitoring.

Communication & Governance

- Weekly meeting (review & planning) giữa các thành viên Mambo và giảng viên hướng dẫn.
- Sprint Review cuối mỗi phase với demo tính năng.
- Thay đổi scope (nếu có) sẽ được ghi nhận qua backlog và ưu tiên trong sprint tiếp theo.

2.4 SECURITY CONSIDERATIONS

1. Access

- Bật **MFA** bắt buộc cho AWS root account và IAM users.
- Sử dụng **IAM Roles** cho EC2/Beanstalk, CodeBuild, thay vì lưu access key trong code.
- Áp dụng nguyên tắc **Least Privilege** cho tất cả IAM policies.

2. Infrastructure

- Tách **Public** và **Private Subnets** trong VPC.
- RDS, ElastiCache, Beanstalk instances nằm trong Private Subnet, không public IP.
- Chỉ ALB và CloudFront public ra ngoài.
- Security Groups chỉ mở port cần thiết (HTTP/HTTPS, DB port giới hạn source).

3. Data

- Bật **Encryption at Rest** cho RDS, S3, và ElastiCache (nếu phù hợp).
- Mã hóa dữ liệu nhạy cảm (mật khẩu user bằng Hashing + Salt ở ứng dụng).
- Không lưu plaintext password/secret trong code hoặc repo.

4. Detection

- Bật **AWS CloudTrail** cho toàn account để audit API calls.
- Sử dụng **AWS CloudWatch Logs & Metrics** cho Beanstalk, RDS, và ALB để theo dõi health.
- (Tùy chọn) Config rules cơ bản để monitor việc cấu hình sai (misconfigurations).

5. Incident Management

- Thiết lập **CloudWatch Alarms** cho CPU, response time, error rate.
- Định nghĩa quy trình phản ứng khi có sự cố: scale up, rollback deploy, restore DB từ backup (PITR).
- Testing quy trình restore RDS từ snapshot định kỳ (trong phạm vi demo/học thuật).

3 ACTIVITIES AND DELIVERABLES

3.1 ACTIVITIES AND DELIVERABLES

| Project Phase | Timeline | Activities | Deliverables/Milestones | Total man-day |
|--------------------------|-----------|--|---|---------------|
| Assessment & Foundation | Week 1–4 | <ul style="list-style-type: none">Thu thập yêu cầuThiết kế kiến trúcSetup VPC | <ul style="list-style-type: none">Solution Architecture Doc, AWS Network, GitHub Repo | 32 Hours |
| Core Feature Development | Week 5–8 | <ul style="list-style-type: none">Code Backend/FrontendUnit TestBusiness Logic | <ul style="list-style-type: none">Web app chạy local ổn định các tính năng chính | 72 Hours |
| Cloud Prep & Refactoring | Week 9–11 | <ul style="list-style-type: none">Refactor code cho CloudConfig Environment VarDockerizeBuildspec | <ul style="list-style-type: none">Dockerfile, Buildspec.yml, Source code chuẩn Cloud Native | 54 Hours |
| Infrastructure & Go-live | Week 12 | <ul style="list-style-type: none">Setup RDS/Redis/EBConfig SecurityCI/CDUAT | <ul style="list-style-type: none">Production URL, Tài liệu bàn giao | 24 Hours |

Project Phase: Assessment & Foundation

- **Timeline:** Week 1–4
- **Activities:**
 - Thu thập yêu cầu business & kỹ thuật.
 - Thiết kế kiến trúc logic và kiến trúc hạ tầng trên AWS.
 - Thiết lập VPC, Subnets, Security Groups cơ bản.
 - Initialize GitHub repository, tạo .NET solution 3-layer.

- **Deliverables / Milestones:**

- Solution Architecture Document.
- AWS Network (VPC + Subnets) được tạo.
- GitHub repo + base code structure.

Project Phase: Setup Base Infrastructure

- **Timeline:** Week 5–8

- **Activities:**

- Phát triển các Service nghiệp vụ: Quản lý sản phẩm, Giỏ hàng, Đơn hàng, User.
- Tích hợp Entity Framework Core, Repository Pattern, Unit of Work.
- Xây dựng giao diện WebShop (Razor Views/Bootstrap). Viết Unit Tests cho các logic quan trọng.

- **Deliverables:**

- Ứng dụng Web Mini-market chạy ổn định trên môi trường Local (đầy đủ tính năng Functional).
- Kết quả Unit Test đạt độ bao phủ yêu cầu.

Project Phase: Setup Application Components

- **Timeline:** Week 9–11

- **Activities:**

- Refactoring: Chuyển đổi các cấu hình cứng (Hard-coded) sang biến môi trường (Environment Variables) để tương thích Elastic Beanstalk.
- Containerization: Viết Dockerfile tối ưu (Multi-stage build) và buildspec.yml.
- S3 Integration: Viết service upload ảnh lên S3 thay vì lưu local.
- Redis Integration: Cấu hình Distributed Cache sử dụng Redis cho Session.

- **Deliverables:**

- Source code đã được tối ưu cho Cloud (Cloud-Native ready).
- File cấu hình Docker và CI/CD (Buildspec) hoàn chỉnh.

Project Phase: AWS Integration & Optimization

- **Timeline:** Week 12

- **Activities:**

- Khởi tạo toàn bộ tài nguyên: RDS (SQL Server), ElastiCache (Redis), Elastic Beanstalk.
- Cấu hình Bảo mật: CloudFront (HTTPS), WAF (Firewall).
- Kích hoạt CI/CD Pipeline (CodePipeline) để deploy tự động.
- Thực hiện UAT (User Acceptance Testing) và kiểm tra CloudWatch Metrics.

- **Deliverables:**

- Hệ thống Production đang chạy (Live Demo URL).
- Tài liệu bàn giao và hướng dẫn vận hành.

3.2 OUT OF SCOPE

Tích hợp thực tế với công thanh toán Việt Nam (VNPay, MoMo, ZaloPay, v.v.) – chỉ mô phỏng.

3.3 PATH TO PRODUCTION

Bản POC/MVP sẽ được xây dựng với các use case chính: đăng ký, đăng nhập, duyệt sản phẩm, giỏ hàng, thanh toán, quản lý đơn hàng.

POC sẽ chạy trong một môi trường **Staging** trên **Elastic Beanstalk** với đầy đủ RDS, S3, ElastiCache.

Để lên Production, các bước bổ sung gồm:

- Tinh chỉnh Auto Scaling rules, RDS instance size, backup & retention.
- Tối ưu WAF rule sets, rate-limiting, CORS, HTTPS enforcement.
- Mở rộng test-cases (load test, security test).

Production setup trong phạm vi dự án sẽ là một **environment demo** với domain Route 53, CloudFront, WAF, sẵn sàng để scale up nếu triển khai thực tế.

4 EXPECTED AWS COST BREAKDOWN BY SERVICES

[AWS Pricing Calculator](#)

Assumptions for cost estimation:

- Sử dụng các instance nhỏ (t3a.small, t3.micro, t4g.micro) phù hợp với business scale mini-market.
- Traffic ở mức vừa phải (dự án demo, không phải hệ thống phục vụ hàng triệu user).
- Tận dụng **AWS Free Tier** trong 12 tháng đầu cho EC2, RDS, S3, CloudFront, ElastiCache.
- Không tính chi phí support plan Enterprise / Business (chỉ dùng Developer/Basic).

5 TEAM

Partner Project Team (Mambo – FPTU)

| Name | Title | Role | Email / Contact Info |
|-----------------|-------|-----------|----------------------|
| Dương Tuấn Kiệt | | Team Lead | |

| | | | |
|----------------------|--------------------------|--|--|
| | Backend Engineer (.NET) | Phát triển backend .NET, business logic, APIs | |
| Hồ Chí Kiệt | Backend Engineer (.NET) | Phát triển backend .NET, business logic, APIs | |
| Nguyễn Hoàng Gia Huy | Cloud & Backend Engineer | AWS architecture & overall system design | |
| Lâm Vĩnh Cường | Cloud Engineer | Thiết kế & triển khai hệ tầng AWS (VPC, RDS, EB, CI/CD) | |
| Nguyễn Đăng Khôi | Frontend Engineer | Thiết kế & triển khai UI/UX , Razor Views, Bootstrap | |

6 RESOURCES & COST ESTIMATES

| Resource | Responsibility | Rate (USD) / Hour |
|-----------------------------|----------------------------|-------------------|
| Solution Architects [1] | Thiết kế kiến trúc AWS | \$30/hr |
| Engineers [3] | Phát triển, test, vận hành | \$20/hr |
| Other (Project Manager) [1] | Quản lý tiến độ, báo cáo | \$25/hr |

| Project Phase | Solution Architects | Engineers | Other (Project Manager) | Total Hours |
|--------------------------|---------------------|-----------|-------------------------|-------------|
| Assessment & Foundation | 8 | 20 | 4 | 32 |
| Core Feature Development | 6 | 60 | 6 | 72 |

| | | | | |
|--------------------------|-------|--------|-------|--------|
| Cloud Prep & Refactoring | 10 | 40 | 4 | 54 |
| Infrastructure & Go-live | 4 | 16 | 4 | 24 |
| Total Hours | 28Hr | 136Hr | 18Hr | 182Hr |
| Total Cost | 840\$ | 2720\$ | 450\$ | 4010\$ |

Cost Contribution distribution between Partner, Customer, AWS:

| Party | Contribution (USD) | % Contribution of Total |
|----------|---------------------------|-------------------------|
| Customer | 0\$ | 0% |
| Partner | 4,010\$ (effort value) | 100% |
| AWS | 200\$ credits (Free Tier) | |

7 ACCEPTANCE

Acceptance process (*mô phỏng theo chuẩn SOW*):

- Sau khi hoàn thành mỗi Phase, nhóm sẽ bàn giao các Deliverables tương ứng cho Customer (Chủ mini-market già lập và giảng viên).
- Customer sẽ có **08 ngày làm việc** để review và test các Deliverables so với tiêu chí chấp nhận đã định nghĩa (success criteria & scope).
- Nếu Deliverable **đạt yêu cầu**, Customer sẽ xác nhận chấp nhận (Acceptance) qua email/biên bản.
- Nếu Deliverable **không đạt yêu cầu**, Customer sẽ phản hồi kèm theo mô tả chi tiết các lỗi hoặc điểm chưa đạt. Nhóm sẽ:
 - Sửa lỗi / điều chỉnh Deliverable trong thời gian hợp lý.
 - Bàn giao lại phiên bản đã sửa để Customer review lại, nhưng review chỉ tập trung vào các lỗi/diểm đã nêu trước đó.
- Nếu Customer không phản hồi trong Acceptance Period, Deliverable được xem là “**deemed accepted**”.