5.2.3. Phân tích rủi ro sản phẩm (Product Risk Analysis) 🔍

Mục tiêu: Nhận biết rủi ro sản phẩm → tập trung nỗ lực kiểm thử → giảm thiểu rủi ro còn lại (residual risk). Nên bắt đầu **càng sớm càng tốt** trong vòng đời phát triển (SDLC).

Phân tích rủi ro sản phẩm gồm **hai bước chính**:

Bước 1: Nhận diện rủi ro (Risk Identification)

Tạo ra **danh sách rủi ro toàn diện** bằng các kỹ thuật:

Kỹ thuật	Mô tả	Ví dụ
Động não (Brainstorming)	Nhóm cùng liệt kê mọi rủi ro có thể nghĩ ra	Team QA + Dev + BA ngồi lại liệt kê: "module nào dễ lỗi nhất?"
Hội thảo (Workshops)	Buổi làm việc có cấu trúc với nhiều bên liên quan	Workshop với khách hàng để xác định tính năng nào quan trọng nhất
Phỏng vấn (Interviews)	Hỏi trực tiếp chuyên gia hoặc stakeholder	Phỏng vấn DBA: "database hiện tại có rủi ro gì về hiệu năng?"
Biểu đồ nguyên nhân - kết quả (Cause-effect diagram)	Sơ đồ xương cá phân tích nguyên nhân gốc	Vẽ fishbone diagram cho vấn đề "thời gian phản hồi chậm"

Bước 2: Đánh giá rủi ro (Risk Assessment)

Sau khi nhận diện → **phân loại, xác định mức độ, ưu tiên** các rủi ro.

Hai cách tiếp cận đánh giá:

Cách tiếp cận	Cách làm	Ví dụ
Định lượng (Quantitative)	Mức độ rủi ro = Xác suất × Tác động (dùng con số)	Xác suất = 0.7, Tác động = 8 → Mức độ = 5.6
Định tính (Qualitative)	Dùng ma trận rủi ro (risk matrix) với mức Cao/Trung bình/Thấp	Xác suất: Cao × Tác động: Cao → Mức độ: **Nghiêm trọng**

> 💡 ***Tại sao cần phân loại?*** *Vì các rủi ro cùng danh mục thường có thể* ***giảm thiểu bằng cách tiếp cận tương tự****. Ví dụ: các rủi ro về bảo mật đều cần penetration testing.*

Kết quả phân tích rủi ro dùng để làm gì?

Ứng dụng	Ví dụ cụ thể
Xác định **phạm vi kiểm thử** (test scope)	Rủi ro thấp → test ít; rủi ro cao → test kỹ
Xác định **mức độ kiểm thử** và **loại hình kiểm thử** (test levels & types)	Rủi ro bảo mật → cần security testing ở mức system test
Chọn **kỹ thuật kiểm thử** và **độ bao phủ** (coverage)	Rủi ro cao → dùng kỹ thuật kỹ hơn như decision coverage thay vì chỉ statement coverage

Ước lượng **nỗ lực kiểm thử** (test effort)	Module thanh toán rủi ro cao → phân bổ 40% effort tổng
Ưu tiên kiểm thử (test prioritization)	Test module thanh toán **trước**, test trang "About Us" **sau**
Xác định **hoạt động bổ sung** ngoài kiểm thử	Code review, pair programming để giảm lỗi từ gốc

---

5.2.4. Kiểm soát rủi ro sản phẩm (Product Risk Control) 🚦

Sau khi phân tích xong → thực hiện các biện pháp **ứng phó**. Gồm hai hoạt động:

Hoạt động	Mục đích	Ví dụ
Giảm thiểu rủi ro (Risk mitigation)	Thực hiện hành động để **giảm mức độ rủi ro**	Viết thêm test case cho module thanh toán, thực hiện code review
Giám sát rủi ro (Risk monitoring)	Theo dõi hiệu quả, cập nhật đánh giá, phát hiện **rủi ro mới**	Sprint review phát hiện: tích hợp API mới phát sinh rủi ro chưa có trong danh sách

Các phương án ứng phó rủi ro (Risk Response Options)

Phương án	Ý nghĩa	Ví dụ
Giảm thiểu (Mitigate)	Kiểm thử để giảm khả năng hoặc tác động	Test kỹ module thanh toán với nhiều kịch bản
Chấp nhận (Accept)	Biết rủi ro nhưng quyết định không hành động	Lỗi UI nhỏ trên trình duyệt ít người dùng → chấp nhận
Chuyển giao (Transfer)	Chuyển rủi ro cho bên khác	Mua bảo hiểm, thuê bên thứ ba chuyên security audit
Lập kế hoạch dự phòng (Contingency plan)	Chuẩn bị sẵn phương án nếu rủi ro xảy ra	Nếu server sập → có hệ thống backup tự động chuyển đổi

Các hành động giảm thiểu rủi ro bằng kiểm thử

Hành động	Ví dụ
Chọn tester có **kinh nghiệm phù hợp**	Rủi ro bảo mật → giao cho tester có chứng chỉ security
Áp dụng **mức độ độc lập** kiểm thử phù hợp	Rủi ro cao → team test độc lập (không phải dev tự test)
Thực hiện **đánh giá** (reviews) và **phân tích tĩnh** (static analysis)	Review code module thanh toán trước khi test động
Áp dụng **kỹ thuật kiểm thử** và **độ bao phủ** phù hợp	Rủi ro cao → boundary value analysis + 100% branch coverage

Áp dụng **loại hình kiểm thử** phù hợp	Rủi ro hiệu năng → performance testing; rủi ro UX → usability testing
Thực hiện **kiểm thử động** + **kiểm thử hồi quy**	Sau mỗi bản fix → chạy regression test để đảm bảo không phát sinh lỗi mới

---

> 🔗 ***Tổng kết luồng quản lý rủi ro sản phẩm: Nhận diện*** → ***Đánh giá*** *(phân loại, xác định mức độ, ưu tiên)* → ***Giảm thiểu*** *(kiểm thử, review...)* → ***Giám sát*** *(theo dõi, cập nhật)* → *quay lại đánh giá nếu có rủi ro mới.*

Một số ghi chú

- Tài liệu dịch từ sách gốc ISTQB Foudation v4.0.1 sang tiếng Việt, nhằm mục đích giảm rào cản tiếp cận các kiến thức về testing tới cộng đồng tester Việt Nam nói chung và các anh chị em muốn tìm hiểu về testing nói riêng
- Tài liệu cố gắng dịch nhiều nhất các từ tiếng Việt để bạn đọc không phải tra từ điển trong quá trình đọc (VD: defects dịch là khuyết tật phần mềm).
- Dự án phi lợi nhuận, bạn có thể thoải mái sử dụng bản dịch, chia sẻ và sửa đổi nếu cần thiết.
- Có góp ý cho dự án, bạn có thể submit góp ý qua link này nha: https://go.betterbytesvn.com/sharing-documentation-feedback.
- Bản dịch được thực hiện bởi tác giả Đỗ Minh Phong. Bạn có thể gửi lời cảm ơn/feedback/ donate tới tác giả thông qua các hình thức:
  - Đăng ký kênh Youtube: https://www.youtube.com/@hoctest_com
  - Follow Fanpage: https://www.facebook.com/hoctest/
  - Tham gia group Playwright Việt Nam trên Facebook: https://www.facebook.com/groups/playwright.automation.test
  - Donate cho tác giả: https://academy.betterbytesvn.com/donate-cho-chung-minh/
  - Facebook cá nhân tác giả: https://www.facebook.com/dominhphong.18/

Xin chân thành cảm ơn bạn, vì đã quan tâm tới tài liệu ^^.