**INTEGRATION TESTING**

1. **Integration testing là gì?**

Kiểm tra tích hợp (Integration testing) là kiểm thử sự tương tác giữa các module, chức năng với nhau trong hệ thống.

1. **Mục đích**

* Đảm bảo tính tương thích và tương tác giữa các thành phần trong hệ thống
* Xác định lỗi và vấn đề khi các thành phần hoạt động cùng nhau
* Đảm bảo hệ thống hoạt động một cách đáng tin cậy và đúng đắn.

1. **Khi nào cần sử dụng Integration testing?**

* Khi kết hợp các module: sau khi kiểm thử các module riêng lẻ,thì sẽ tiến hành kiểm thử tích hợp các module với nhau để kiểm tra sự tương tác và tương thích của chúng khi kết hợp lại.
* Kiểm tra giao diện và luồng thông tin: đảm bảo tính tương thích giữa giao diện và các luồng thông tin giữa các thành phần trong hệ thống.

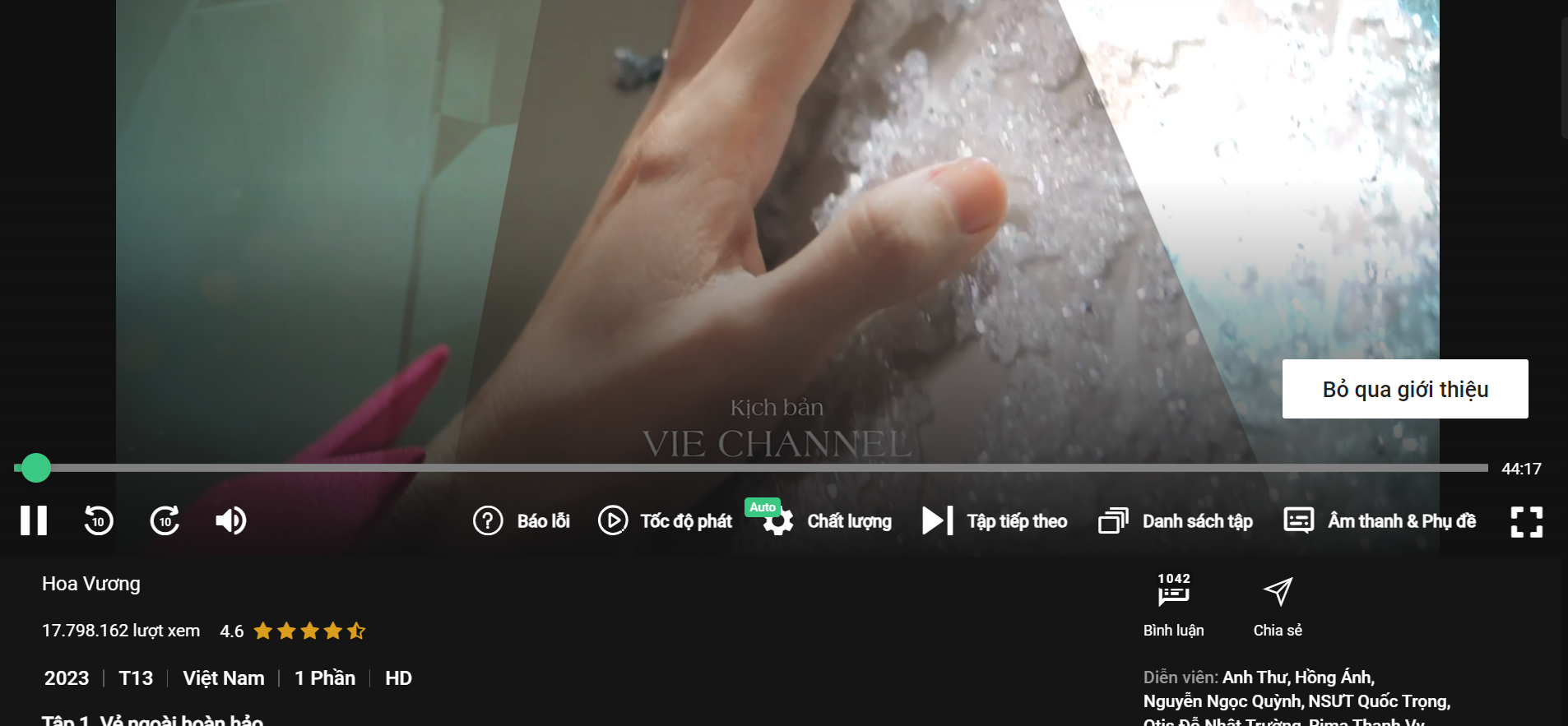
1. **Cấp độ Integration testing**

* Tích hợp thành phần: tập trung vào sự tương tác và giao diện giữa các thành phần trong hệ thống.Kiểm thử tích hợp thành phần được thực hiện sau khi kiểm thử thành phần.

Ví dụ: Xây dựng chức năng quản lí nhân viên, cần tích hợp các module: hàm nhập thông tin nhân viên, hàm thêm nhân viên, hàm xoá nhân viên, hàm sửa thông tin nhân viên,hàm tính lương.

* Tích hợp hệ thống: tập trung vào sự tương tác giữa các hệ thống.Kiểm thử tích hợp hệ thống được thực hiện sau khi kiểm thử hệ thống.

Ví dụ: Vào web VieOn, chọn xem một tập phim, mỗi tập sẽ có phần đánh giá phim và bình luận. Tích hợp watch, đánh giá, bình luận được xem là tích hợp hệ thống.



1. **Quy trình Integration testing**

* Chuẩn bị kế hoạch
* Viết kịch bản kiểm thử, test case
* Tiến hành kiểm thử
* Báo cáo lỗi (nếu có)
* Theo dõi và retest.

1. **Phương pháp tiếp cận**
   1. **Big-bang**

* Các thành phần được tích hợp cùng lúc, sau đó tiến hành kiểm thử.
* Ưu điểm: đơn giản, nhanh chóng, thích hợp cho dự án nhỏ
* Nhược điểm: khó xác định nguyên nhân gốc rễ của lỗi, bỏ qua các lỗi tích hợp phức tạp,…

Ví dụ: Thực hiện từng chức năng như Đăng ký , Đăng Nhập, Xem thông tin cá nhân, sau đó tích hợp các chức năng này để xác nhận sự tương tác giữa các chức năng đó.

* 1. **Bottom-up**
* Kiểm thử từ dưới lên, theo thứ tự từ các thành phần nhỏ đến các thành phần lớn
* Ưu điểm : Phát hiện lỗi sớm
* Nhược điểm: Khó khăn trong việc thực hiện sớm

Ví dụ: Thực hiện từng tính năng thêm, xoá,sửa.Rồi đó gán cho chức năng xem danh sách sinh viên.Sau đó gán chức năng xem danh sách sinh viên cho chức năng xem thông tin sinh viên của giảng viên

* 1. **Top- Down**
* Kiểm thử từ trên xuống, theo thứ tự từ các thành phần lớn đến các thành phần nhỏ
* Ưu điểm: các module quan trọng được ưu tiên, giúp phát hiện những lỗi nghiêm trọng từ sớm,…
* Nhược điểm: cần nhiều stub, các module ở mức thấp hơn ít được kiểm thử đầy đủ

Ví dụ: Muốn thêm một món ăn vào danh sách món ăn nhưng chức năng xem danh sách món ăn chưa hoàn thành, trường hợp này ta cần stub để giả lập chức năng xem danh sách món ăn.

1. **Các lỗi thường gặp**

* Đối với kiểm thử tích hợp thành phần:

+ Dữ liệu không chính xác, bị thiếu, mất dữ liệu

+ Giao diện không khớp

+ Trình tự hoặc thời gian gọi giao diện không chính xác

* Đối với kiểm thử tích hợp hệ thống:

+ Dữ liệu không chính xác, bị thiếu, mất dữ liệu

+ Giao diện không khớp

+ Lỗi giao tiếp giữa các hệ thống

+ Không tuân thủ các quy định bảo mật bắt buộc

1. Ưu điểm và nhược điểm của Integration testing

* Ưu điểm:

+ Phát hiện sớm các lỗi khi tích hợp các module với nhau

+ Kiểm tra sự tương tác : đảm bảo các chức năng hoạt động cùng nhau một cách chính xác

+ Tăng tính ổn định và tin cậy

* Nhược điểm:

+Khó khăn trong việc định vị lỗi

+ Phụ thuộc vào các chức năng: cần tích hợp các chức năng để kiểm tra nhưng có một chức năng chưa hoàn thành có thể gây ra sự trì hoãn.

+ Phức tạp, tốn nhiều thời gian