

5.1.6. Kim tự tháp kiểm thử (Test Pyramid)

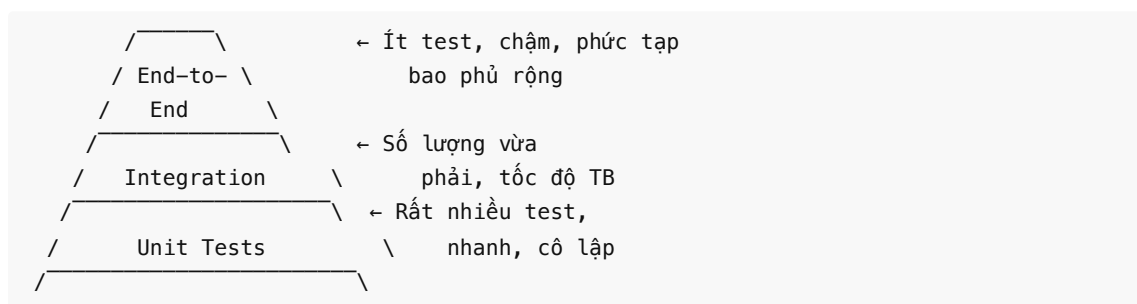
Khái niệm

Kim tự tháp kiểm thử là mô hình minh họa cách các bài kiểm thử có **độ chi tiết (granularity)** khác nhau, giúp đội ngũ:

- Phân bổ nỗ lực kiểm thử hợp lý
- Định hướng chiến lược **tự động hóa kiểm thử (test automation)**

👉 Ý tưởng cốt lõi: **Nhiều test nhỏ-nhanh ở đáy, ít test lớn-chậm ở đỉnh.**

Cấu trúc kim tự tháp



Đặc điểm	Tầng đáy (Bottom)	Tầng đỉnh (Top)
Độ chi tiết (Granularity)	Cao — test từng phần nhỏ	Thấp — test luồng tổng thể
Mức cô lập (Isolation)	Cao — ít phụ thuộc	Thấp — phụ thuộc nhiều thành phần
Tốc độ	Nhanh ⚡	Chậm 🐢
Số lượng test	Nhiều	Ít
Ví dụ	Test hàm <code> tínhThue(amount) </code> trả đúng giá trị	Test toàn bộ luồng: đăng nhập → chọn hàng → thanh toán → nhận email xác nhận

Mối quan hệ thứ bậc

Càng lên cao trong kim tự tháp:

- Độ chi tiết **giảm**
- Mức cô lập **giảm** (phụ thuộc nhiều thành phần hơn)
- Thời gian thực thi **tăng**
- Số lượng test cần thiết **giảm** (mỗi test đã bao phủ phạm vi rộng)

Các biến thể phổ biến

Mô hình	Tầng đáy	Tầng giữa	Tầng đỉnh
---------	----------	-----------	-----------

Cohn (2009) — bản gốc	Unit Tests	Service Tests	UI Tests
Biến thể phổ biến	Unit (Component) Tests	Integration (Component Integration) Tests	End-to-End Tests

Ngoài ra, các cấp độ kiểm thử khác (xem mục 2.2.1) cũng có thể được áp dụng vào mô hình này tùy theo dự án.

Ví dụ thực tế — Ứng dụng thương mại điện tử

Tầng	Test gì?	Ví dụ cụ thể	Số lượng
Unit	Hàm/method đơn lẻ	<code> tinhPhiShip(weight, distance)</code> trả đúng giá trị	~500 tests
Integration	Tương tác giữa các module	API POST <code>/orders</code> ghi đúng vào database và gọi đúng service tồn kho	~100 tests
End-to-End	Luồng nghiệp vụ hoàn chỉnh	User đăng nhập → thêm sản phẩm → thanh toán VNPAY → nhận email xác nhận	~20 tests

Tóm tắt nhanh

Đáy: Nhiều + Nhanh + Cô lập → Unit tests (nền tảng vững chắc)
Giữa: Vừa phải → Integration tests (kết nối đúng)
Đỉnh: Ít + Chậm + Toàn diện → E2E tests (hệ thống hoạt động đúng)

🚩 **Nguyên tắc vàng:** Nếu kim tự tháp bị "lật ngược" (quá nhiều E2E, quá ít unit test) → bộ test sẽ chậm, khó bảo trì và phản hồi trễ khi có lỗi.

Một số ghi chú

- Tài liệu dịch từ sách gốc ISTQB Foundation 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 ^^.