



1

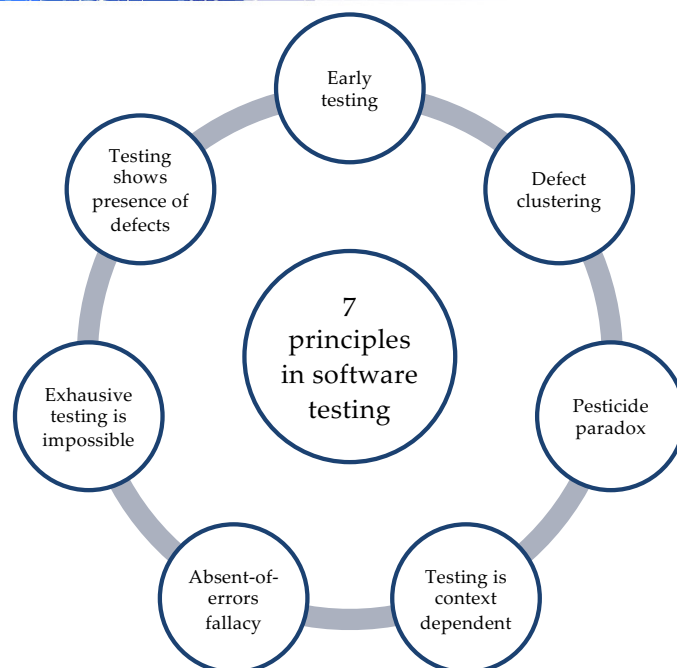


2

Tổng quan kiểm thử phần mềm

- Kiểm thử là các hoạt động thực thi chương trình nhằm tìm ra lỗi của chương trình.
- Kiểm thử nghĩa là thực thi chương trình với dữ liệu được chuẩn bị trước.
- Kiểm thử phần mềm thường chia làm hai loại chính là kiểm thử thủ công (manual testing) và kiểm thử tự động (automation testing).

Các nguyên tắc kiểm thử





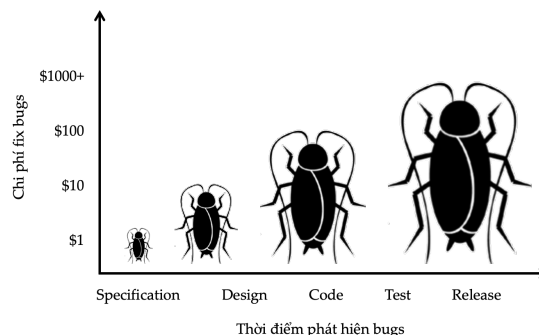
Các nguyên tắc kiểm thử

- Kiểm thử cho thấy sự có mặt của lỗi (testing shows presence of defects): mục đích chính của kiểm thử là phát hiện ra lỗi, không nhằm chứng minh phần mềm không có lỗi.
- Kiểm thử vét cạn là chuyện không thể (exhaustive testing is impossible).



Các nguyên tắc kiểm thử

- Kiểm thử sớm (early testing): quá trình kiểm thử nên bắt đầu càng sớm càng tốt trong quy trình phát triển phần mềm vì chi phí phải trả sẽ càng tăng nếu lỗi được tìm thấy càng trễ trong quy trình phát triển phần mềm





Các nguyên tắc kiểm thử

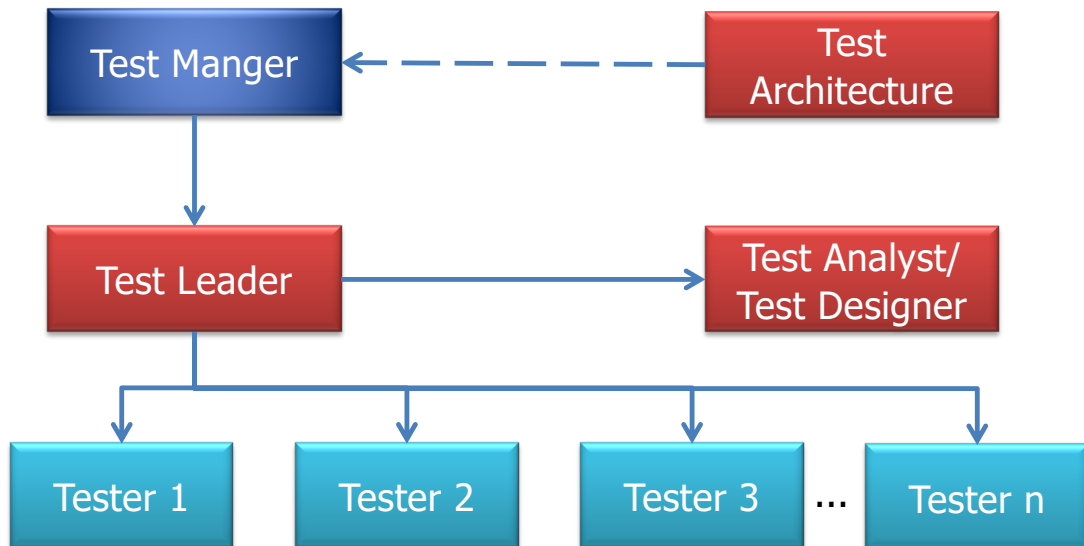
- Gom cụm lỗi (defect clustering): lỗi thường tập trung ở một vài module, tính năng của hệ thống.
- Nguyên lý thuốc trừ sâu (Pesticide paradox): một bộ test case nếu được thực hiện lặp đi lặp lại sẽ khó phát hiện bug mới.



Các nguyên tắc kiểm thử

- Kiểm thử phụ thuộc ngữ cảnh (Testing is context dependent): mỗi lĩnh vực có yêu cầu, chức năng, ràng buộc, rủi ro và mục đích kiểm thử khác nhau.
- Sự vắng mặt của lỗi là ảo tưởng (Absence-of-errors fallacy): việc tin rằng tìm được một số lượng lớn các lỗi của hệ thống và giải quyết hết tất cả các lỗi này thì hệ thống sẽ hoạt động thành công là ảo tưởng.

Các vai trò kiểm thử



Các vai trò kiểm thử



How people reacts differently to a single word.

"Bug"



Tester

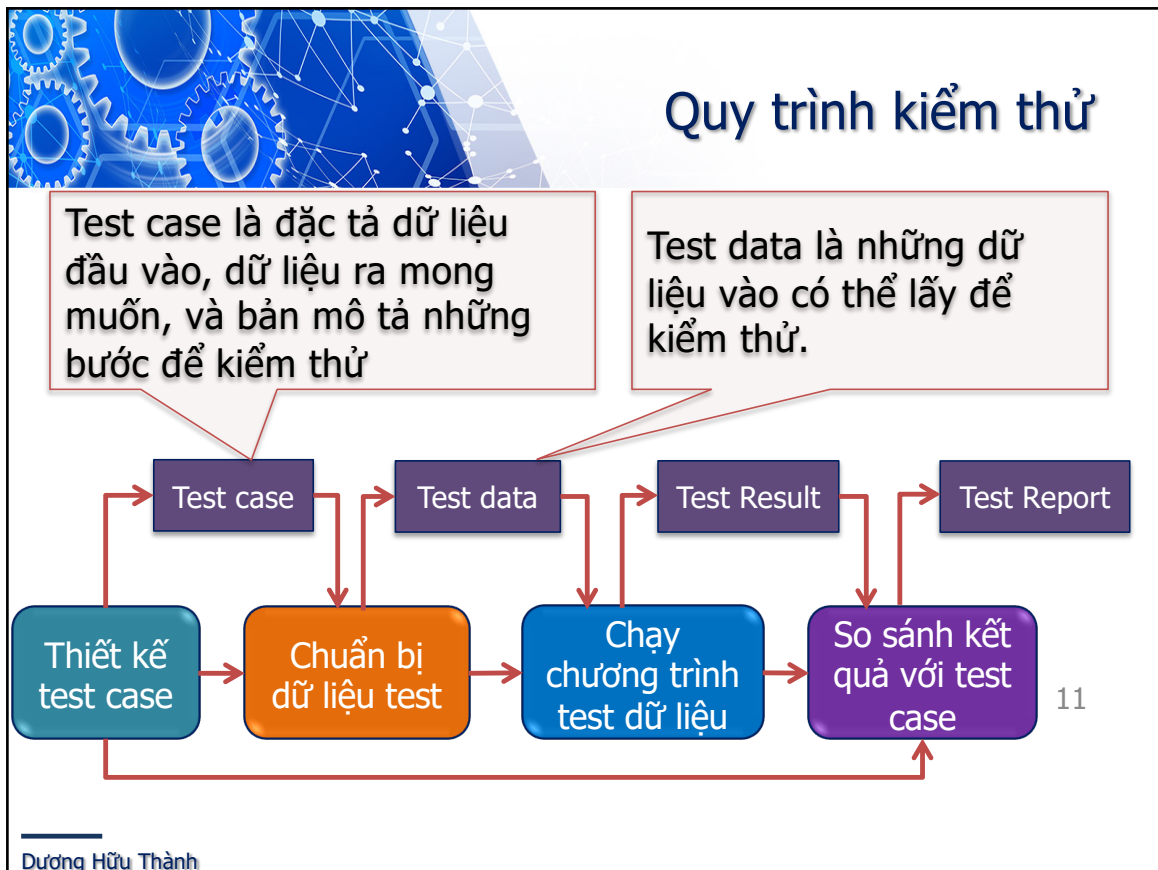


Developer

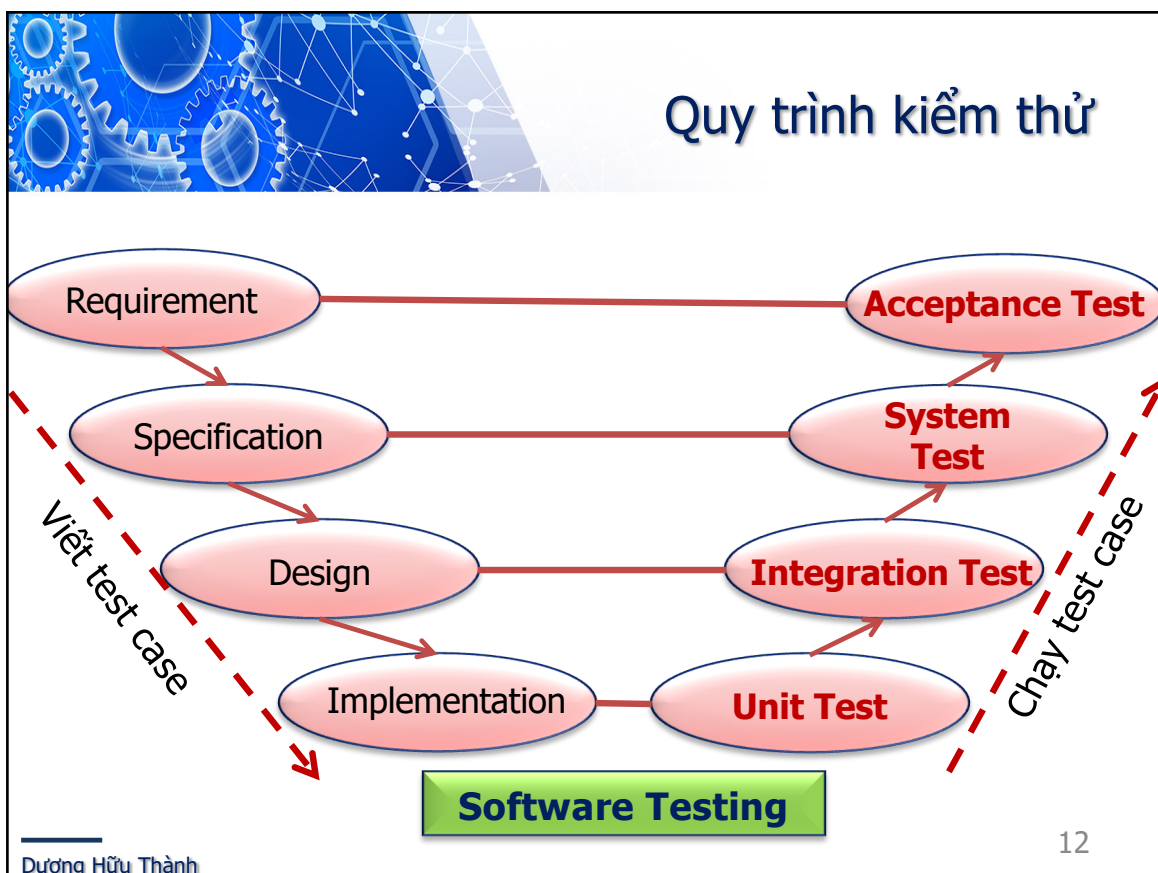


Manager

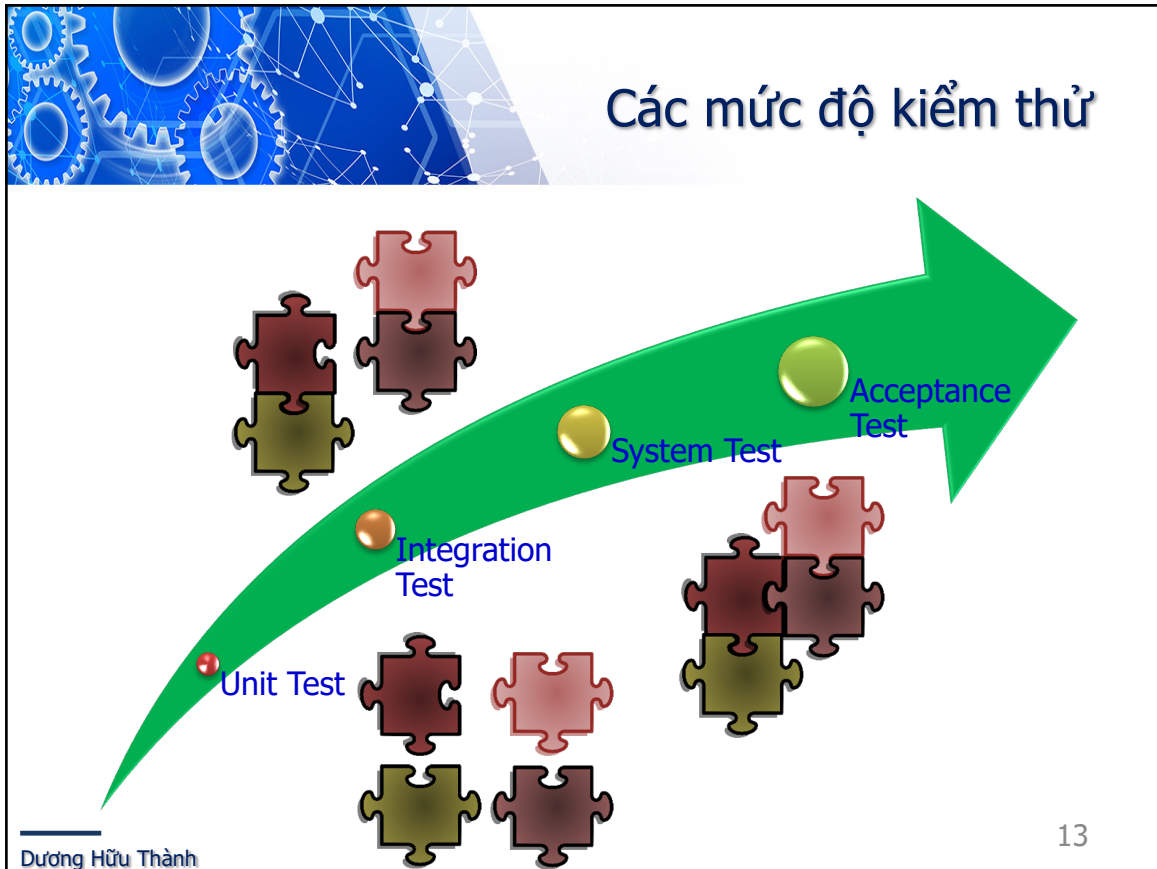
Nguồn: internet



11



12



13

Unit Test

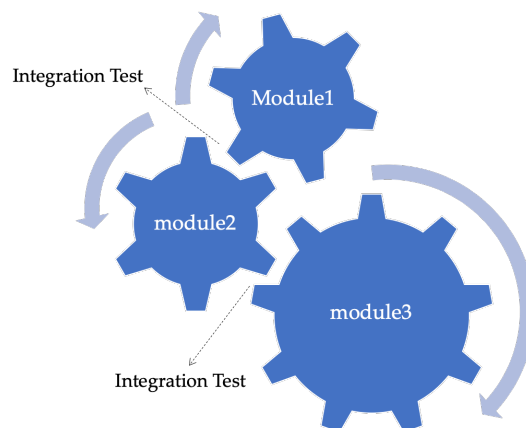
- Một đơn vị (unit) là một thành phần nhỏ nhất của thể kiểm thử được, nó có thể là một hàm, một thủ tục, một phương thức hoặc một lớp.
- Các đơn vị thường được phát triển độc lập, việc tích hợp được thực hiện ở các giai đoạn sau đó.
- Kiểm thử đơn vị (unit test) dùng kiểm thử trên từng đơn vị riêng rẽ, thường được thực hiện bởi chính lập trình viên phát triển đơn vị đó.

Dương Hữu Thành

14

Integration Test

- Kiểm thử tích hợp (Integration Test) sẽ được thực hiện sau khi kết hợp các đơn vị, thành phần riêng rẽ với nhau và tập trung kiểm thử **sự tương tác** nhằm phát hiện lỗi.



System Test

- Kiểm thử hệ thống (System Test) thực hiện kiểm thử trên hệ thống đầy đủ sau khi các thành phần đã được tích hợp nhằm đảm bảo hệ thống tuân thủ các đặc tả yêu cầu của người dùng.
- Kiểm thử hệ thống được thực hiện bởi một nhóm độc lập trong quy trình phát triển, điều này giúp việc đánh giá hệ thống khách quan hơn.



Acceptance Test

- Kiểm thử chấp nhận (Acceptance Test) được thực hiện chính bởi khách hàng nhằm cung cấp tính tin cậy hệ thống, đảm bảo sản phẩm hoạt động đúng những gì khách hàng mong đợi.
 - Alpha Testing: người dùng kiểm thử phần mềm ngay trong môi trường phát triển phần mềm.
 - Beta Testing: người dùng kiểm thử phần mềm trong môi trường thực của người dùng, phần mềm được công khai cho người dùng quan tâm dùng thử



Kiểm thử tĩnh

- Kiểm thử tĩnh là các kỹ thuật kiểm thử không cần thực thi mã nguồn chương trình.
- Hai kỹ thuật được sử dụng thường xuyên trong kiểm thử tĩnh là đánh giá (review) và phân tích tĩnh (static analysis).



Kỹ thuật review

- Review để tìm và gỡ bỏ các lỗi, những điều chưa rõ trong các tài liệu trước khi chúng được sử dụng trong quy trình phát triển.
- Có hai loại review là formal review và informal review.
 - formal review được thực hiện theo quy trình chính thức, có kế hoạch.
 - informal review khi thực hiện không xác định trước các bước, quy trình thực hiện.

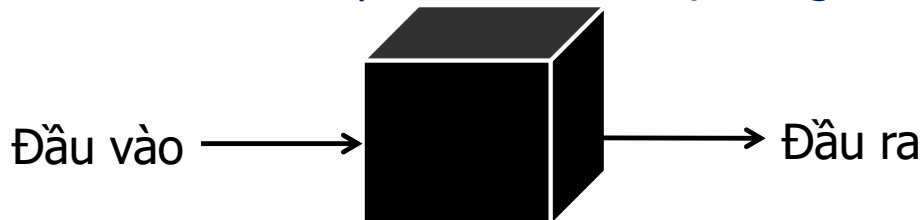


Kỹ thuật review

- Dù thuộc loại nào, hầu như đều có tối thiểu các việc cơ bản:
 - Các tài liệu cần review sẽ được nghiên cứu bởi reviewer và xác định các vấn đề.
 - reviewer thông báo đến tác giả bằng lời hoặc bằng văn bản.
 - Sau đó các tác giả có phản hồi lại nhận xét của các reviewer và cập nhật tài liệu phù hợp.

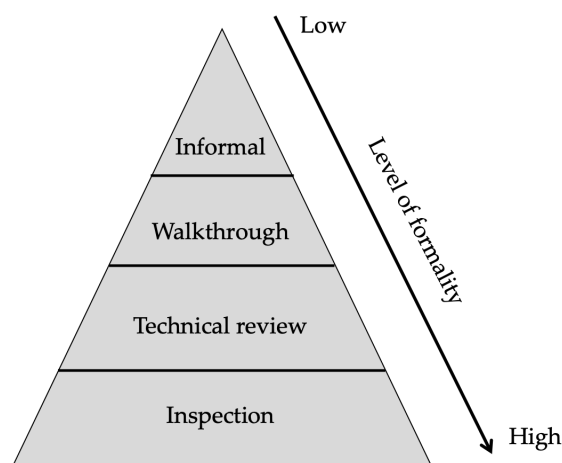
Kỹ thuật review

- **Kỹ thuật kiểm thử hộp đen** (black-box testing) là kỹ thuật kiểm thử các chức năng của hệ thống dựa vào bản đặc tả yêu.
- Trong kiểm thử hộp đen, kiểm thử viên không cần quan tâm chi tiết hiện thực bên trong của hệ thống, mà chỉ cần tập trung vào các đầu vào và kiểm tra kết quả đầu ra của hệ thống.



Kỹ thuật review

- Một số kỹ thuật review





Kỹ thuật phân tích tĩnh

- Kỹ thuật thứ hai là phân tích tĩnh (static analysis) cho phép mã nguồn được phân tích các vấn đề cấu trúc chương trình hoặc những điểm yếu của chương trình (source code), các mô hình phần mềm (model) một cách hệ thống mà không cần thực thi chương trình.



Kỹ thuật phân tích tĩnh

- Phân tích tĩnh giúp phát hiện các vấn đề sớm, đưa ra những cảnh báo những điểm đáng nghi ngờ trong mã nguồn chương trình hoặc trong thiết kế bằng cách tính toán một số độ đo (metrics) như độ phức tạp chương trình.
- Nó cũng giúp cải thiện khả năng bảo trì mã nguồn và các thiết kế của phần mềm.



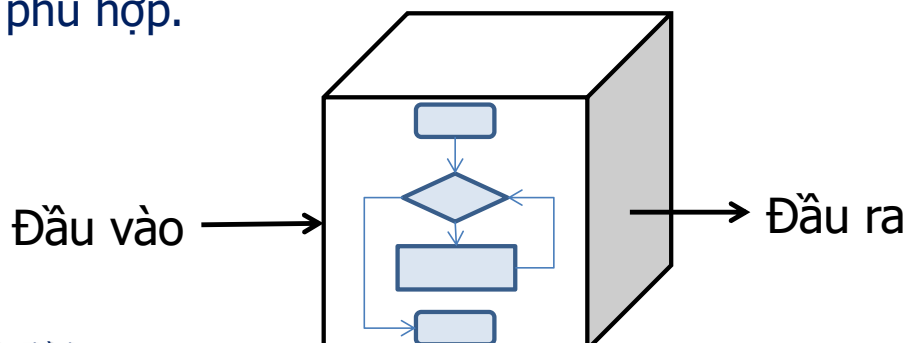
Kỹ thuật phân tích tĩnh

- Các vấn đề có thể phát hiện được:
 - Tham chiếu đến biến nhưng giá trị của nó chưa được chỉ định (undefined).
 - Biến khai báo nhưng không được sử dụng.
 - Phát hiện các hàm và thủ tục không được gọi.
 - Vi phạm chuẩn lập trình.
 - Chương trình thiếu tính an toàn (security).
 - Sử dụng các cú pháp không chính xác trong ngôn ngữ lập trình hay một ngôn ngữ thiết kế phần mềm đang sử dụng.



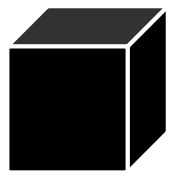
Kiểm thử động

- **Kỹ thuật kiểm thử hộp trắng** (white-box testing) là kỹ thuật kiểm thử yêu cầu biết logic và cấu trúc mã nguồn bên trong hệ thống. Kiểm thử viên cần biết rõ công việc bên trong mã nguồn, làm rõ từng đơn vị hoặc mã nguồn nào không phù hợp.



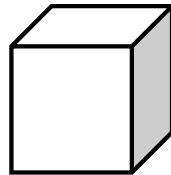
Kiểm thử động

- **Kỹ thuật kiểm thử hộp xám** (gray-box testing) là kỹ thuật kết hợp ưu điểm của kiểm thử hộp đen và kiểm thử hộp trắng, tức là kiểm thử viên yêu cầu có chút hiểu biết công việc bên trong của hệ thống.



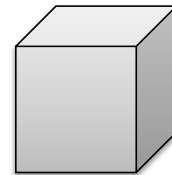
Black-box

+



White-box

=



Gray-box

Q&A