

# QUẢN LÝ DỰ ÁN PHẦN MỀM

---

## Bài 10: Tích hợp và kiểm thử

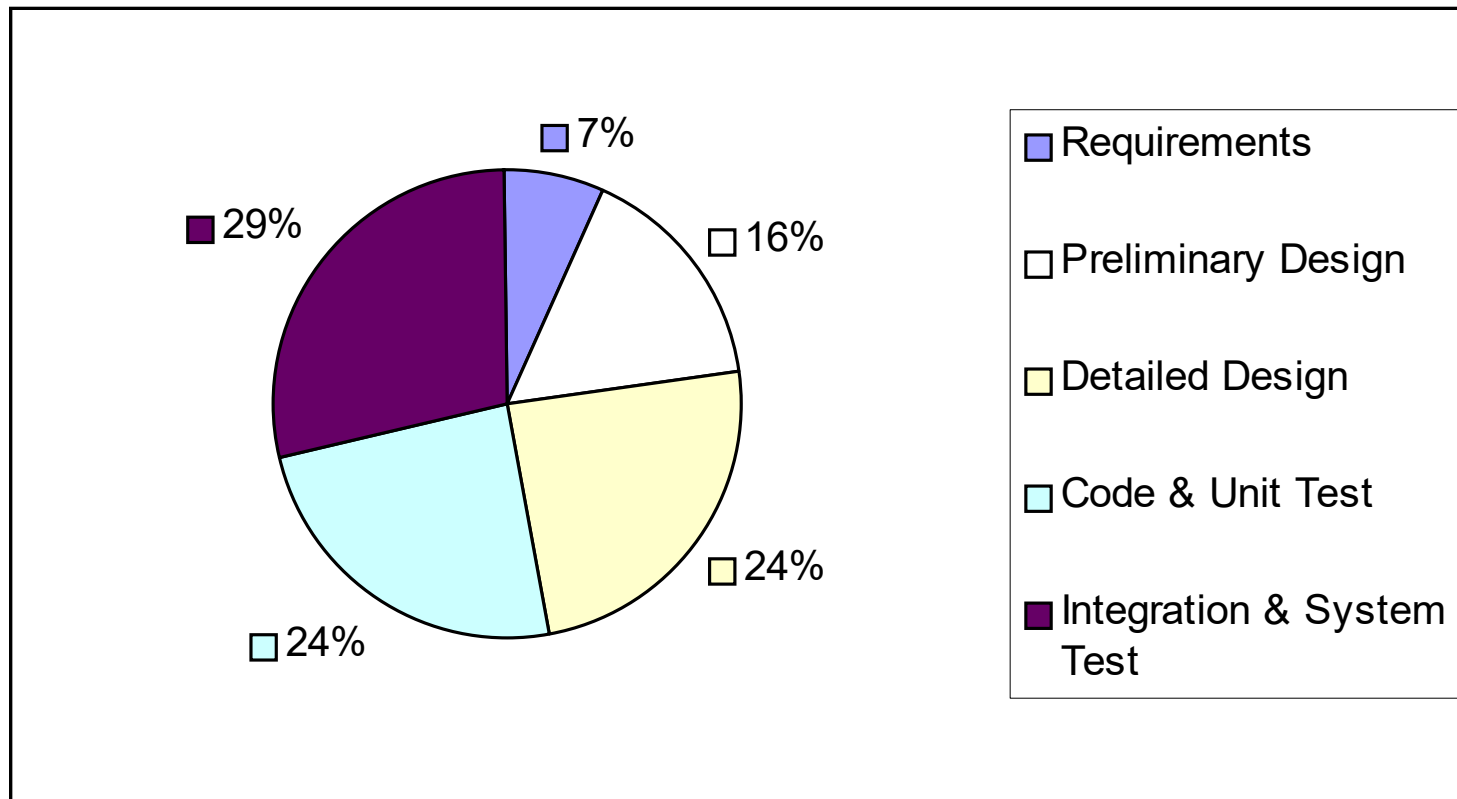
# Nội dung bài học

---

- Đảm bảo chất lượng phần mềm
- Tích hợp
- Lập kế hoạch kiểm thử
- Các loại kiểm thử
- Đơn vị đo cho kiểm thử
- Các công cụ kiểm thử

# Các chi phí phát triển dự án<sup>2</sup>

---



# Tích hợp và kiểm thử

---

- Phát triển/Tích hợp/Kiểm thử
  - Nơi chung nhất cho sự chồng chéo lịch thực hiện và các hoạt động
- Đôi khi coi Tích hợp/Kiểm thử là cùng một quá trình
- Tổng hợp dần dần các tính năng
- Đội đảm bảo chất lượng dự án làm việc song song với đội phát triển dự án

# Các cách tiếp cận cho tích hợp

---

- Trên xuống
  - Các hệ thống cơ bản hoặc điều hành được cài đặt trước tiên
  - Được kết hợp vào hệ thống tế bào nhỏ nhất
- Dưới lên
  - Bắt đầu với các modul và các hàm cần thiết lập
  - Các đơn vị chương trình riêng biệt (sau khi được kiểm thử chức năng) được kết hợp vào các hệ thống con
  - Các hệ thống con được kết hợp lại với nhau thành hệ thống tổng thể

# Tích hợp

---

- Ai thực hiện kiểm thử tích hợp?
  - Có thể là người phát triển hệ thống hoặc/và đội đảm bảo chất lượng
- Điều phối Nhân viên và ngân sách khá căng thẳng
- Các vấn đề:
  - Căng thẳng
  - Ngày bàn giao sản phẩm đang đến gần
  - Các lỗi không mong đợi
  - Các vấn đề về động cơ thúc đẩy
  - Các xung đột về tính chấp nhận sản phẩm của người sử dụng

# Xác định và kiểm tra tính đúng đắn

---

- V & V
- Xác định tính đúng đắn của sản phẩm
  - Chúng ta đang tạo ra đúng sản phẩm cần thiết?
- Kiểm tra sự đúng đắn tính năng của sản phẩm
  - Chúng ta đang xây dựng sản phẩm đúng?
  - Kiểm thử
  - Điều tra
  - Phân tích tĩnh

# Đảm bảo chất lượng

---

- QA hoặc SQA (Đảm bảo chất lượng phần mềm)
- Đảm bảo tốt chất lượng xuất phát từ quá trình tốt
- Khi nào SQA bắt đầu?
  - Ngay trong quá trình tìm hiểu yêu cầu
- Một chức năng của chuẩn CMM cấp 2
- QA là cửa sổ nhòm vào dự án của bạn



# Các kế hoạch kiểm thử

---

- Kế hoạch đảm bảo chất lượng phần mềm
  - Nên hoàn thành bản kế hoạch này ở thời điểm gần cuối giai đoạn xác định yêu cầu
- Xem ví dụ
  - Thậm chí sử dụng chuẩn IEEE 730

# Quá trình đảm bảo chất lượng phần mềm

---

- Các phần điển hình
  - Mục đích
  - Các tài liệu tham khảo tới
  - Quản lý
  - Tài liệu
  - Các chuẩn, thực tế, quy định chuẩn, đơn vị đo
    - Đo đặc chất lượng
    - Thực tập kiểm thử

# Quá trình đảm bảo chất lượng phần mềm

---

- Các phần điển hình (tiếp theo)
  - Xem xét và kiểm tra
    - Các quá trình và công việc xem xét lại cụ thể là
      - Xem xét lại tài liệu yêu cầu
      - Xem xét lại kế hoạch kiểm thử
      - Xem xét lại mã nguồn
      - Xem xét lại khi kết thúc dự án
  - Quản lý rủi ro
    - gắn việc đảm bảo chất lượng với kế hoạch quản lý rủi ro
  - Báo cáo các trục trặc và hành động sửa đúng
  - Các công cụ, kỹ thuật, phương pháp luận
  - Thu thập các bản ghi và dự đoán

# Chất lượng phần mềm

---

- Tính lần vết được (Traceability)
  - Khả năng dò tìm các mối quan hệ giữa các sản phẩm công việc
  - Ví dụ: các yêu cầu/thiết kế/trường hợp kiểm thử tốt thế nào
- Xem xét lại dự án chính thức
  - Được thực hiện tại thời điểm kết thúc mỗi giai đoạn
  - bản yêu cầu dự án....

# Kiểm thử

---

- Chạy thử chương trình với những đầu vào xác định trước
- So sánh các kết quả thực tế với kết quả mong đợi
- Kiểm thử là một dạng của lấy mẫu
- Không thể chứng minh là không xuất hiện lỗi
- Tất cả các phần mềm đều có lỗi
- Kiểm thử không phải là quá trình gỡ rối chương trình

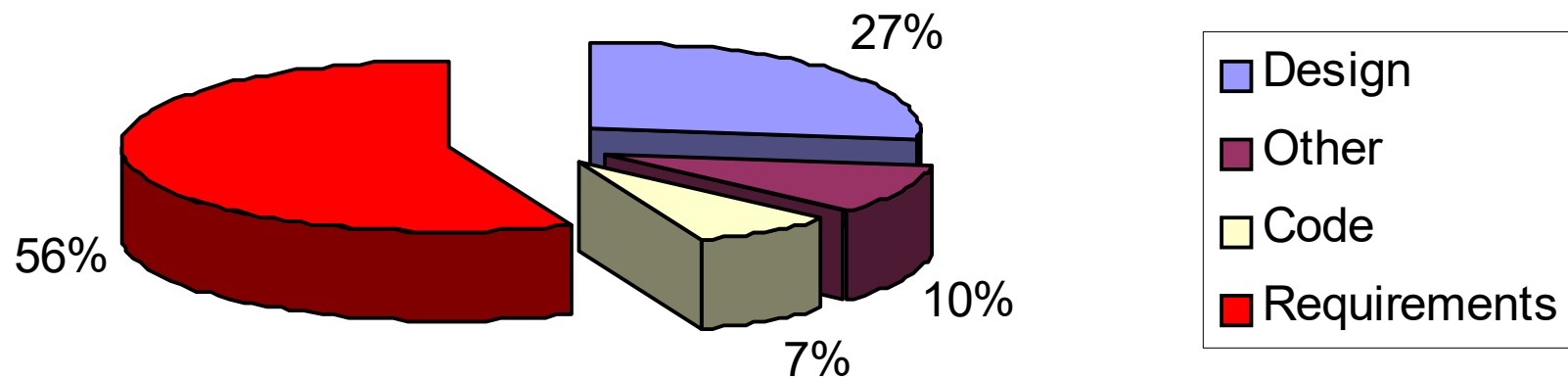
# Các trường hợp kiểm thử

---

- Yếu tố quan trọng của một kế hoạch kiểm thử
- Có thể bao gồm các đoạn mã, dữ liệu, bảng liệt kê các việc
- Có thể chiếu tới một ma trận quản lý tất cả các yêu cầu (Requirements Coverage Matrix)
  - Một công cụ dò tìm vết

# Nguồn gốc gây lỗi

---



# Luồng kiểm thử dự án

---

- Kiểm thử đơn vị chức năng
- Kiểm thử tích hợp
- Kiểm thử hệ thống
- Kiểm thử sự chấp nhận của người dùng



# Kiểm thử hộp đen

---

- Kiểm thử chức năng
- Chương trình là một "hộp đen"
  - Không quan tâm tới nó làm việc như thế nào mà tới nó làm cái gì
  - Tập trung vào đầu vào và đầu ra
- Các trường hợp kiểm thử dựa trên bản phát biểu các yêu cầu của dự án

# Kiểm thử hộp trắng

---

- Chứa cấu trúc của chương trình
- Bao gồm
  - Các lệnh được thực hiện
  - Các luồng đường đi dựa trên mã nguồn

# Kiểm thử đơn vị chức năng

---

- hay còn gọi kiểm thử modul
- Một loại của kiểm thử hộp trắng
  - Đôi khi được coi như hộp đen
- Ai thực hiện kiểm thử đơn vị chức năng?
  - Những người phát triển hệ thống
  - Việc kiểm thử đơn vị chức năng được viết bằng mã nguồn
    - Cùng ngôn ngữ với các hàm chức năng
    - Còn được gọi là "công cụ chỉ dẫn kiểm thử" (Test drivers)
- Khi nào thực hiện kiểm thử chức năng?
  - cùng trong quá trình phát triển hệ thống
  - Khi từng modul riêng biệt được hoàn thành

# Kiểm thử đơn vị chức năng

---

- Các kiểm thử có thể được gộp nhóm với nhau
  - Bộ kiểm thử
- JUnit
  - Là một phần trong phương pháp XP
  - Lập trình kiểm tra trước

# Kiểm thử tích hợp

---

- Kiểm thử giao diện giữa các thành phần cấu thành hệ thống
- Bước đầu tiên sau kiểm thử đơn vị chức năng
- Các cấu thành có thể làm việc một mình nhưng có thể không chạy được đúng khi được ghép lại với nhau
- Lỗi có thể tồn tại trong một môđun nhưng lại thể hiện ở một môđun khác
- Các kiểm thử loại hộp đen

# Kiểm thử hệ thống

---

- Kiểm thử toàn bộ hệ thống
- Một loại kiểm thử hộp đen

# Kiểm thử sự chấp nhận của người dùng

---

- Mốc cuối cùng trong quá trình kiểm thử
- Khách hàng cuối cùng kiểm tra và ký nhận
- Đôi khi mượn những người nào đó thực hiện kiểm thử bản beta
- Khách hàng chỉ hài lòng khi thoả mãn yêu cầu của họ
- Dựa trên "tiêu chí chấp nhận"
  - Các điều kiện mà phần mềm đáp ứng được để khách hàng chấp nhận hệ thống
  - Diễn hình được xác định trước khi hợp đồng được ký
  - Sử dụng các điều kiện có thể đo lường được, định lượng được

# Kiểm thử nhanh

---

- Chạy lại việc kiểm thử ngay sau khi sửa lỗi hoặc thực hiện những thay đổi đối với phần mềm hoặc môi trường
- Ví dụ: QA tìm thấy lỗi, lập trình viên sửa lỗi, QA chạy kiểm thử nhanh để kiểm tra
- Các công cụ tự động rất hữu ích trong việc này



# Kiểm thử tính tương thích

---

- Kiểm thử trên các môi trường hệ thống nền khác
  - Ví dụ: Kiểm thử đối với nhiều trình duyệt khác nhau
  - Nó có hoạt động đúng với Netscape/IE, Windows/Mac

# Các mốc kiểm thử bên ngoài

---

- bản đầu tiên Alpha, bản thứ hai Beta
- Kiểm thử bởi người dùng ngoài tổ chức
  - Thường được thực hiện bởi các người sử dụng cuối
- Phiên bản Alpha
  - Được đưa cho một số người sử dụng rất hạn chế
  - Sản phẩm chưa hoàn thiện các tính năng
- Trong giai đoạn sau của quá trình kiểm thử
- Phiên bản Beta
  - Kiểm thử và đánh giá của khách hàng
  - Các tính năng quan trọng nhất
  - Tốt nhất là sau khi phần mềm đã chạy ổn định

# Các mốc kiểm thử bên ngoài

---

- Giá trị của kiểm thử bản beta
  - Kiểm thử trong thế giới thực
  - Nhận một bản đánh giá phần mềm
  - Quảng cáo bán hàng
  - Tăng cường nhân viên của bạn
- Đừng cân nhắc các đặc tính dựa trên phiên bản đó
  - Quá muộn!
- Các nhân viên kiểm thử bản Beta phải được thuê từ bên ngoài
  - Từ: nhóm sẵn có, nhóm quảng cáo sản phẩm, hỗ trợ kỹ thuật,
- Đòi hỏi một vai trò "quản lý phiên bản beta"
- Tất cả công việc đó được xếp lịch bởi giám đốc dự án

# Các mốc kiểm thử bên ngoài

---

- Phiên bản có thể phân phối (Release candidate)
  - Được chuyển tới nơi phân phối nếu kiểm thử thành công
- Phân phối tới kho (RTM)
  - Sản phẩm được chính thức chuyển đến kho phân phối
- Đạt tới trạng thái ổn định trước mỗi mốc kiểm thử này
  - Đội dự án tập trung vào chất lượng, tích hợp và tính ổn định

# Các chương trình con kiểm thử

---

- Hai nghĩa
- 1. Một bộ gồm các hướng dẫn từng bước một để chỉ dẫn nhân viên kiểm thử thực hiện công việc
  - Danh sách các hành động và các kết quả mong đợi
- 2. Một chương trình nhỏ dùng cho việc kiểm thử tự động

# Kiểm thử tĩnh

---

- Kiểm tra xem xét lại
  - Hầu hết các chi tiết tỉ mỉ có thể cần xem lại
  - Bản đề xuất, hợp đồng, lịch thực hiện, bản yêu cầu, mã nguồn, mô hình dữ liệu, các kế hoạch kiểm thử
- Kiểm tra theo cặp
  - kiểm tra có phương pháp các sản phẩm phần mềm theo cặp để tìm lỗi và những thay đổi cần thiết
  - Mục đích: loại bỏ lỗi sớm và hiệu quả
  - Lập kế hoạch bởi giám đốc dự án, thực hiện trong các buổi họp và được ghi lại bằng tài liệu
  - Hoạt động theo chuẩn CMM cấp 3

# Kiểm thử tự động

---

- Các nhân viên kiểm thử = không hiệu quả
- Ưu điểm
  - Giảm tổng chi phí cho kiểm thử
  - Các công cụ có thể chạy vắng mặt nhân viên
  - Công cụ chạy các bộ kiểm thử nhanh hơn con người
  - Tốt cho các loại kiểm thử nhanh và tương thích
  - Các kiểm thử tạo một sự tổng hợp tri thức
  - Có thể giảm số lượng cán bộ đảm bảo chất lượng
- Nhược điểm
  - Không phải mọi cái đều có thể tự động được
  - Đường cong học hoặc tri thức trong các công cụ
  - Chi phí cho các công cụ cuối cao \$5-80K

# Các công cụ kiểm thử

---

- Nhận dữ liệu và chạy lại
- Phân tích sự bao phủ
- Kiểm thử năng suất
- Quản lý các trường hợp kiểm thử



# Kiểm thử khả năng tải và chịu tải

---

- Đẩy hệ thống vào trạng thái chịu tải trên mức giới hạn
- Thường thực hiện bằng kiểm thử tự động
  - Bồi đội đảm bảo chất lượng dự án
  - Gần thời gian cuối của giai đoạn kiểm thử chức năng
- Có thể chỉ ra
  - Các vấn đề chức năng còn ẩn giấu
  - Công suất lớn nhất của hệ thống
  - Dữ liệu không chấp nhận được hoặc thiếu dịch vụ
  - Xác định liệu yêu cầu về năng suất đã thoả mãn chưa
    - Nhớ rằng, đây là những yêu cầu phi chức năng

# Kiểm thử khả năng tải và chịu tải

---

- Các đơn vị đo lường
  - Thời gian đáp ứng nhỏ nhất chấp nhận được
  - Số lượng người dùng đồng thời nhỏ nhất chấp nhận được
  - Thời gian chết nhỏ nhất chấp nhận được
- Các nhà cung cấp thứ cấp: cuối mức độ cao
  - Segue
  - Mercury
  - Empirix

# Các đơn vị đo năng suất

Xấu	Tốt
Phải hỗ trợ được 500 người dùng	Phải hỗ trợ được 500 người dùng đồng thời
Thời gian đáp ứng là 10 giây	Trung bình/lớn nhất/90% thời gian đáp ứng là X giây
Phải xử lý được 1 triệu lần đọc trong một ngày	Phải xử lý được tải lớn nhất là 28 trang trong một giây

Source: Athens Consulting Group

# Kiểm thử khác

---

- Kiểm thử việc cài đặt
  - Rất quan trọng nếu không phải hệ thống trang Web
  - Có thể dẫn đến chi phí hỗ trợ cao và sự không hài lòng của khách hàng
- Kiểm thử việc sử dụng
  - Xác định lại sự hài lòng của người dùng
    - Tính định hướng
    - Tính thân thiện người dùng
    - Khả năng hoàn thành các nhiệm vụ chính

# Những thủ thuật

---

- Phân tích Pareto
  - Luật 80-20
    - 80% lỗi từ 20% mã nguồn
  - Tìm ra các modul có vấn đề
- Lay nhiệm giữa các pha
  - Kiểm thử cuối mỗi pha
  - Ngăn chặn các vấn đề lây lan từ pha này sang pha kia
- Burn-in
  - Cho phép các hệ thống chạy trong khoảng thời gian dài hơn
  - Nhiều kiểu kiểm thử chịu tải

# Những lưu ý

---

- “Code Freeze”
  - Khi lập trình viên ngừng viết mã nguồn mới và chỉ thực hiện sửa chữa các lỗi
  - Xảy ra tại một thời điểm thay đổi trong quá trình tích hợp/kiểm thử
- Tỷ lệ nhân viên kiểm thử/nhân viên viết mã
  - Phụ thuộc vào từng dự án
  - Thường là 1:3 hoặc 1:4
  - Kích cỡ nhân viên tăng: cần quản lý QA và/hoặc trưởng nhóm QA sớm

# Ngừng kiểm thử

---

- Khi nào bạn ngừng?
- Hiếm khi tất cả các lỗi được sửa đến khi phân phối sản phẩm
- Tập trung vào tất cả những lỗi quan trọng. mức độ ca và trung bình
- Thường xảy ra khi thời gian đã hết
- Ký nhận bàn giao cuối cùng
  - Bởi: khách hàng, nhân viên kỹ thuật, quản lý sản phẩm

# Các đơn vị kiểm thử

---

- Tải: thời gian đáp ứng lớn nhất chấp nhận được, số người dùng đồng thời nhỏ nhất
- Thảm họa: thời gian hệ thống ngừng hoạt động lớn nhất cho phép
- Tính tương thích: số lượng nhỏ nhất/lớn nhất trình duyệt và hệ điều hành hỗ trợ được
- Tính khả dụng: tỉ lệ thông qua nhỏ nhất từ các nhóm người quan tâm
- Tính chức năng: bao gồm các yêu cầu, 100% thành công khi kiểm thử tự động các bộ sản phẩm



# Đơn vị đo lỗi

---

- Rất quan trọng đối với giám đốc dự án
- Số lượng lỗi nguy hiểm
  - Được phân hạng bởi sự nguy hiểm
    - Rất nguy hiểm, nguy hiểm mức độ cao, trung bình, thấp
- Mở và đóng

# Theo dõi lỗi

---

- Có công cụ làm việc này cho bạn
  - Bugzilla, TestTrack Pro, Rational ClearCase
  - Một vài công cụ tốt miễn phí hoặc phí thấp
- Đảm bảo tất cả các thành viên cần thiết của đội có thể truy nhập được (có nghĩa là gần hết toàn bộ)
- Có các buổi họp xem xét lại lỗi thường xuyên
  - có thể là hàng tuần ngay từ lúc bắt đầu kiểm thử, hàng ngày với giai đoạn tích hợp
- Ai có thể nhập thông tin các lỗi vào hệ thống theo dõi?
  - Nhiều người: nhân viên QA, lập trình viên, người phân tích, quản lý, người dùng, giám đốc dự án

# Theo dõi lỗi

---

- Các trường
  - Trạng thái: mở, đóng, chờ
  - Ngày tạo ra, cập nhật, đóng
  - Mô tả vấn đề
  - Số hiệu phiên bản phân phối
  - Người nhập vào
  - Thứ tự ưu tiên: thấp, trung bình, cao, rất cao
  - Nhận xét: bởi QA, lập trình viên, và thành viên khác

# Đơn vị đo lỗi

---

- Tỉ lệ mở
  - Bao nhiêu lỗi mới sinh ra trong một khoảng thời gian
- Tỉ lệ đóng
  - Bao nhiêu được đóng trong cùng khoảng thời gian đó
  - Ví dụ: 10 lỗi/ngày
- Tỉ lệ thay đổi
  - Số lần cùng một vấn đề được cập nhật
- Đếm số sửa chữa sai
  - Những sửa lỗi mà không thực sự sửa được (vẫn mở)
  - Một đơn vị đo sự dao động trong dự án

# Tỉ lệ lỗi

---

- Nghiên cứu với Microsoft
  - 10-20/KLOC trong quá trình kiểm thử
  - 0.5/KLOC sau khi đưa ra phiên bản

# Môi trường kiểm thử

---

- Bạn cần kiểm thử ở đâu đó. Nơi nào?
- Thường phân ra môi trường phần cứng và môi trường mạng

# Môi trường phần cứng

---

- Phát triển
- QA
- Staging
- Sản phẩm

# Môi trường phần cứng

---

- Môi trường điển hình
  - Phát triển
    - Nơi các lập trình viên làm việc
    - Các kiểm thử đơn vị chức năng thực hiện ở đây
  - Kiểm thử
    - Cho tích hợp, hệ thống và kiểm thử nhanh
  - Stage
    - Cho burn-in và kiểm thử tải
  - Sản phẩm
    - Môi trường triển khai cuối cùng



# Các vai trò QA

---

- Quản lý QA
  - Thuê đội QA; tạo kế hoạch kiểm thử; lựa chọn các công cụ, quản lý đội
  - Lương: \$50-80K/năm, \$50-100/giờ
- Nhân viên kiểm thử/kỹ sư kiểm thử
  - Thực hiện kiểm thử chức năng; viết các chương trình tự động kiểm thử
  - Lương: \$35-70K/năm, \$40-100/giờ
- Quản trị hệ thống
  - Hỗ trợ chức năng QA nhưng không phải là thành viên của đội QA chính thức
- Soạn bản sao chép/Viết tài liệu
  - Hỗ trợ QA; cũng không nằm trong đội chính thức

# Câu hỏi?

---