

Thuộc tính chất lượng

Trần Mạnh Cường

Nội dung

- Thuộc tính chất lượng (Quality attributes)
- Các chiến thuật

Tài liệu tham khảo

- Chương 4~12, Software Architecture in Practice, 3rd Edition
- Chương 16, Microsoft Application Architecture Guide, 2nd Edition

Yêu cầu

- Yêu cầu được lấy, thể hiện qua nhiều dạng: đoạn văn bản, hệ thống có sẵn, ca sử dụng,...
- Yêu cầu có thể được phân làm các loại
 - Yêu cầu chức năng: hệ thống phải làm gì?
 - Yêu cầu thuộc tính chất lượng: thể hiện chất lượng của các chức năng hoặc của cả hệ thống
 - Ràng buộc: có thể xem là các quyết định đã có từ trước, hệ thống phải tuân theo (ví dụ ngôn ngữ lập trình, phải dùng thư viện nào,...)

Mối quan hệ giữa kiến trúc với yêu cầu

- Các yêu cầu chức năng sẽ được thỏa mãn bằng cách gán các trách nhiệm (chức năng) cần thực hiện vào các phần tử của kiến trúc
- Các yêu cầu phi chức năng sẽ được thỏa mãn thông qua các cấu trúc (structures)
- Ràng buộc được thỏa mãn bằng cách chấp nhận các quyết định có trước

Chức năng

- Chức năng: khả năng của hệ thống thực hiện các công việc (what)
- Chức năng KHÔNG quyết định kiến trúc
 - Với một tập các yêu cầu chức năng -> có vô số kiến trúc
 - Việc chia thành lớp, mô đun,... là để phục vụ các yêu cầu khác

Chức năng vs. thuộc tính chất lượng

- Chức năng: khi người dùng click vào “Sign up”, form đăng ký sẽ mở ra
- Thuộc tính chất lượng
 - Sau bao lâu
 - Form có dễ dùng không
 - Số lần form không truy cập được
- Ranh giới giữa chức năng và thuộc tính chất lượng không phải lúc nào cũng rõ ràng

Thuộc tính chất lượng

- Availability
- Conceptual Integrity
- Interoperability
- Maintainability
- Manageability
- Performance
- Reliability
- Reusability
- Scalability
- Security
- Supportability
- Testability
- User Experience / Usability
- ...

Thuộc tính chất lượng

- Không có định nghĩa thống nhất
- Chồng chéo



Phân loại

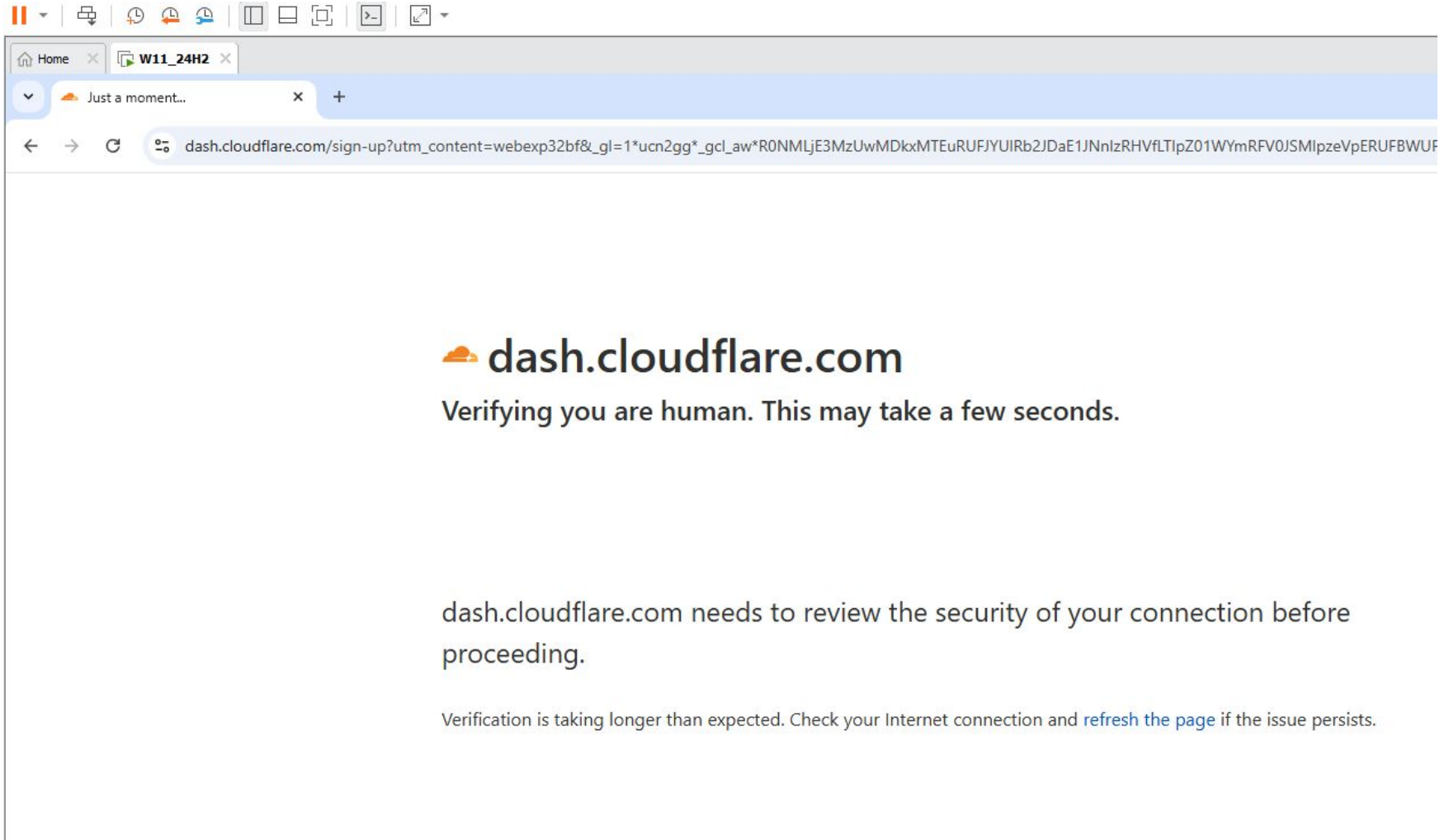
- Thuộc tính trong thời gian chạy
- Thuộc tính trong thời gian phát triển
- Thuộc tính hệ thống
- Thuộc tính người dùng

Đánh đổi



Tham khảo thêm: <https://msdn.microsoft.com/en-us/library/bb402962.aspx>

Đánh đổi



Khả năng sẵn sàng

- Thể hiện sự có mặt, có thể phục vụ khi cần
- Lỗi: fault, error, failure
- $A = \text{MTBF} / (\text{MTBF} + \text{MTTR})$
 - MTBF: mean time between failures
 - MTTR: mean time to repair

Availability = 99%, tương ứng với thời gian không phục vụ 3 ngày 15.6 giờ/năm

Availability = 99.99%, tương ứng với thời gian không phục vụ 52 phút 34 giây/năm

Khả năng sẵn sàng

3/2024: Cổ phiếu của Meta (công ty mẹ Facebook) giảm gần 2% ngay sau sự cố. Vốn hóa công ty này giảm 20 tỷ USD. Khối tài sản của Mark Zuckerberg mất gần 3 tỷ USD, xuống còn khoảng 172 tỷ USD.


Chiến thuật cho tính sẵn sàng


- Phát hiện lỗi
- Phục hồi sau khi bị lỗi
- Chống lỗi

Chiến thuật phát hiện lỗi

- *Monitor* – Xây dựng một thành phần có chức năng quan sát trạng thái của các phần khác nhau trong hệ thống
- *Ping/echo* – gửi một thông điệp và nhận thông điệp phản hồi (sau một khoảng thời gian định trước)
- *Heartbeat* – một thành phần đều đặn gửi thông điệp (heartbeat) đến một thành phần khác
- *Time stamp* – gán thời gian (số thứ tự) cho các sự kiện (chủ yếu là thông điệp trong hệ thống phân tán) nhằm phát hiện lỗi trong thứ tự của các sự kiện


Chiến thuật phát hiện lỗi









**IsItDown**
RightNow?

 Is It Down Right Now ?

"Is It Down Right Now" monitors the status of your favorite web sites and checks whether they are down or not. Check a website status easily by using the below test tool. Just enter the url and a fresh site status test will be performed on the domain name in real time using our online website checker tool. For detailed information, check response time graph and user comments.

Enter a domain below to check whether it is down or not...

 Top Websites

Netflix  Netflix.com is up. Checked 0 seconds ago.	Facebook  Facebook.com is up. Checked 0 seconds ago.
Youtube  Youtube.com is up. Checked 2 mins ago.	Yahoo Mail  Mail.yahoo.com is up. Checked 30 mins ago.
Google  Google.com is up. Checked 32 mins ago.	Outlook  Outlook.com is up. Checked 1 hour 22 mins ago.
Steam Community  Steamcommunity.com is up. Checked 1 min ago.	Yahoo  Yahoo.com is up. Checked 2 mins ago.

Các chiến thuật phục hồi

- *Active redundancy (hot restart)* – tất cả các thành phần đều đáp ứng cho một sự kiện, chỉ duy nhất đáp ứng (kết quả) của một thành phần được sử dụng (thường là kết quả trả về sớm nhất)
- *Passive redundancy (warm restart/dual redundancy/triple redundancy)* – một thành phần (chính) sẽ đáp ứng sự kiện và thông báo cho các thành phần còn lại (phụ) cập nhật trạng thái;
- *Spare (cold spare)* – một thành phần (thường là máy tính, hệ thống) có cấu hình tương tự, không hoạt động, chỉ khi thành phần chính có lỗi mới được khởi động
- *Rollback* – quay ngược hệ thống lại trạng thái trước khi lỗi xảy ra

Chiến thuật chống lỗi

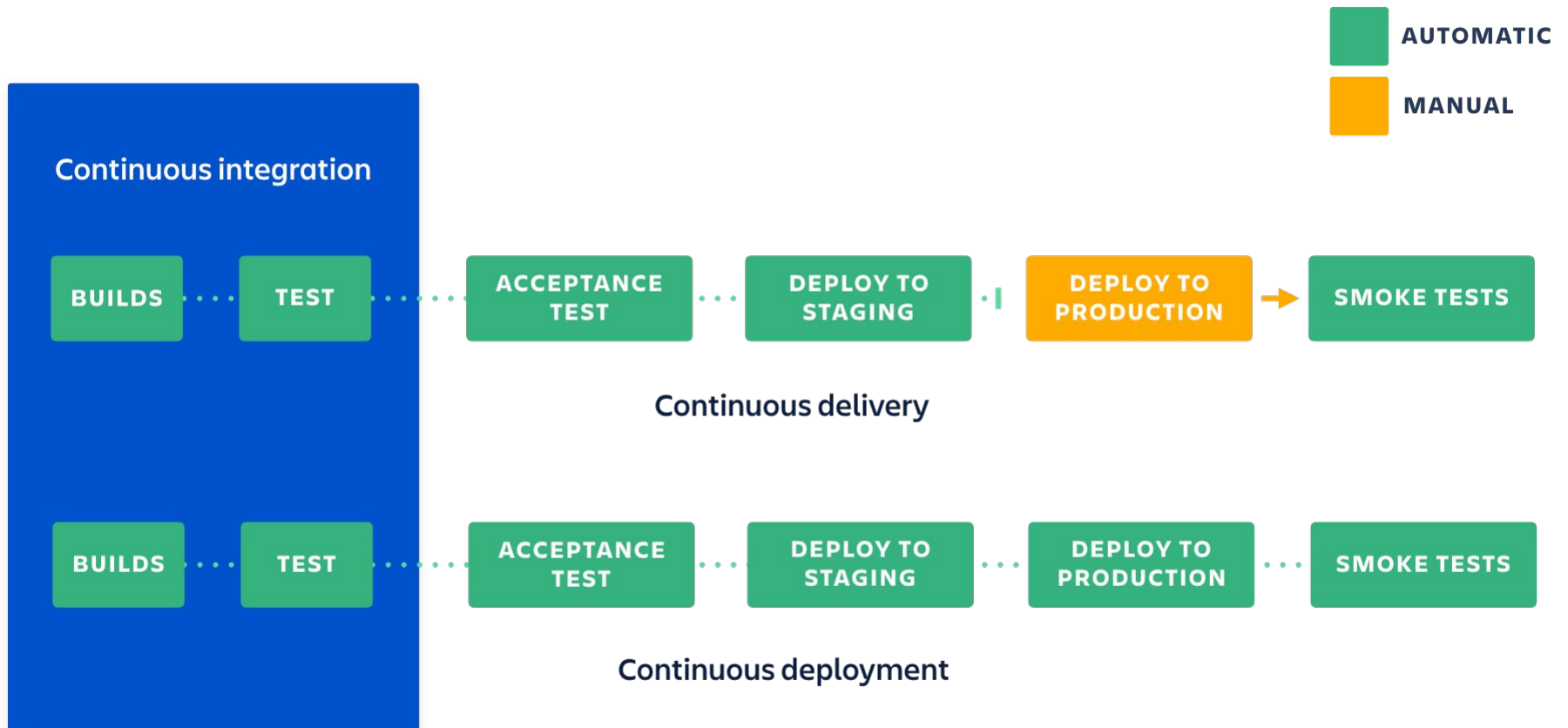
- *Voting* – sử dụng nhiều thành phần để thực hiện cùng một việc, kết quả sẽ được so sánh và chọn giá trị phù hợp nhất
 - *Replication* – một dạng của voting, các thành phần hoàn toàn giống nhau (không tránh được lỗi của thiết kế và lập trình)
 - *Functional redundancy* – một dạng của voting, các thành phần được thiết kế và cài đặt khác nhau (cho ra cùng kết quả)
 - *Analytic redundancy* – các thành phần được thiết kế, cài đặt khác nhau; các input và output cũng khác nhau

Chiến thuật chống lỗi

- *Tạm dừng* – Tạm thời dừng sử dụng một thành phần trong hệ thống và thực hiện các thao tác để chống lỗi (như reset)
- *Giao dịch (transactions)* – Gom một số các hoạt động liên quan thành giao dịch để tránh trường hợp hệ thống rơi vào trạng thái lỗi
- *Mô hình dự đoán*: quan sát trạng thái của hệ thống, trong trường hợp thấy có khả năng sẽ dẫn đến lỗi, thực hiện các thao tác cần thiết

Khả năng triển khai

- Ngày nay, khoảng thời gian giữa hai lần triển khai (deploy) phần mềm ngày càng được rút ngắn
- JIT: just in time (feature release, bug fixes)
- Quá trình triển khai: chuyển code thành hệ thống có thể sử dụng bởi người dùng cuối
- Triển khai liên tục (continuous deployment): quá trình triển khai hoàn toàn tự động
- Chuyển giao liên tục (continuous delivery): quá trình triển khai tự động cho đến bước chuyển phần mềm vào môi trường sản phẩm (production)



- **Các môi trường:**

- **Phát triển:** nơi các thành phần phần mềm sẽ được phát triển và kiểm thử đơn vị. Sau khi vượt qua kiểm thử và review, code sẽ được đưa lên hệ thống quản lý phiên bản để thực hiện quá trình “build”
- **Tích hợp:** biên dịch các đoạn code mới được cập nhật, kết hợp với các thành phần khác để tạo ra bản chạy được. Kiểm thử đơn vị và kiểm thử tích hợp sẽ được tiến hành
- **Staging:** là môi trường kiểm thử nhiều khía cạnh khác nhau của hệ thống: hiệu năng, an ninh,...
- **Production:** triển khai và quan sát

- Dễ triển khai: thuộc tính của phần mềm cho biết phần mềm có thể triển khai trong một khoảng thời gian và nỗ lực biết trước
- Lựa chọn kiến trúc sẽ ảnh hưởng đến khả năng triển khai
- Kiến trúc sư phần mềm cần biết một bản thực thi của phần mềm được cập nhật thế nào
 - Được push hay pull đến máy thực thi
 - Được tích hợp với hệ thống hiện có như thế nào

Một số chiến thuật

- **Triển khai từng phần:** Thay vì gửi bản mới cho tất cả người dùng, chỉ gửi cho một bộ phận
- **Quay ngược (roll back):** phải có cơ chế cho phép quay trở về phiên bản cũ nếu phiên bản triển khai có lỗi
- **Sử dụng script cho việc triển khai:** script triển khai phải được quan tâm như một chương trình và được kiểm thử, quản lý phiên bản

- Quản lý tương tác của các dịch vụ: một dịch vụ có thể có nhiều phiên bản (mới, cũ); yêu cầu từ client có thể được chuyển đến phiên bản phù hợp
- Triển khai các phần tử phụ thuộc cùng nhau

Khả năng tích hợp (Integrability)

- Là khả năng một thành phần/phần mềm có thể tương tác để cùng hoạt động với các thành phần/hệ thống khác
- Muốn tăng cường tính tích hợp khi
 - Hệ thống của chúng ta có thể sẽ cung cấp chức năng cho các hệ thống không biết trước
 - Hệ thống của chúng ta sử dụng các chức năng cung cấp bởi các hệ thống khác

Chiến thuật cho khả năng tích hợp

- Đóng gói (encapsulate)
- Sử dụng thành phần trung gian
- Tuân thủ theo các chuẩn phổ biến
- Trừu tượng hóa các *dịch vụ* chung

Khả năng thay đổi

- Thể hiện khả năng hệ thống có thể thay thế, chỉnh sửa một vài thành phần
- Mục tiêu
 - Điều khiển sự phức tạp của việc thay đổi
 - Giảm thời gian và chi phí thay đổi
- Khái niệm coupling và cohesion
 - 2 mô đun có tính coupling cao sẽ dẫn đến sự phụ thuộc cao, thay đổi một mô đun sẽ ảnh hưởng đến mô đun còn lại
 - Các chức năng trong một thành phần có tính cohesion càng cao thì càng tốt

Chiến thuật cho tính thay đổi

- Giảm kích thước các mô đun
- Tăng cohesion của các chức năng trong từng mô đun
- Giảm coupling giữa các mô đun
 - Đóng gói
 - Sử dụng thành phần trung gian
 - Trừu tượng hóa các dịch vụ chung
- Kết nối (binding) muộn
 - Phương pháp giao tiếp
 - Đa hình (OOP)

Hiệu năng

- Thể hiện khả năng xử lý của hệ thống trong một đơn vị thời gian
- Phân biệt
 - Response time (thời gian đáp ứng): thời gian xử lý 1 yêu cầu
 - Throughput (thông lượng?): số lượng yêu cầu xử lý trong một đơn vị thời gian (giây)

Chiến thuật cho hiệu năng

- Quản lý yêu cầu (request): queue, priority
- Giảm overhead: giảm việc chuẩn hóa dữ liệu (để tăng tính hợp tác, đầu vào có thể ở nhiều dạng khác nhau), giảm thời gian truyền nhận, giảm thời gian đọc/ghi dữ liệu
- Tăng khả năng xử lý đồng thời
 - Tăng tài nguyên tính toán
 - Duy trì nhiều bản sao dữ liệu (giảm thời gian truy xuất)

An ninh

- **Các khía cạnh của an ninh**
 - Bảo mật (confidentiality)
 - Toàn vẹn (integrity)
 - Sẵn sàng (availability)
 - Xác thực (authentication)
 - Không chối từ (nonrepudiation)
 - Phân/kiểm quyền (authorization)

Các chiến thuật

- Phát hiện tấn công
 - Phát hiện xâm nhập
 - Phát hiện từ chối dịch vụ
 - Kiểm tra toàn vẹn thông điệp
 - Phát hiện chậm trễ thông điệp
- Chống tấn công
 - Xác thực người dùng
 - Kiểm quyền người dùng
 - Hạn chế truy cập (dùng DMZ)
 - Mã hóa dữ liệu

Tổng kết

- Giới thiệu thuộc tính chất lượng
- Một số chiến thuật để đạt được yêu cầu về thuộc tính chất lượng

