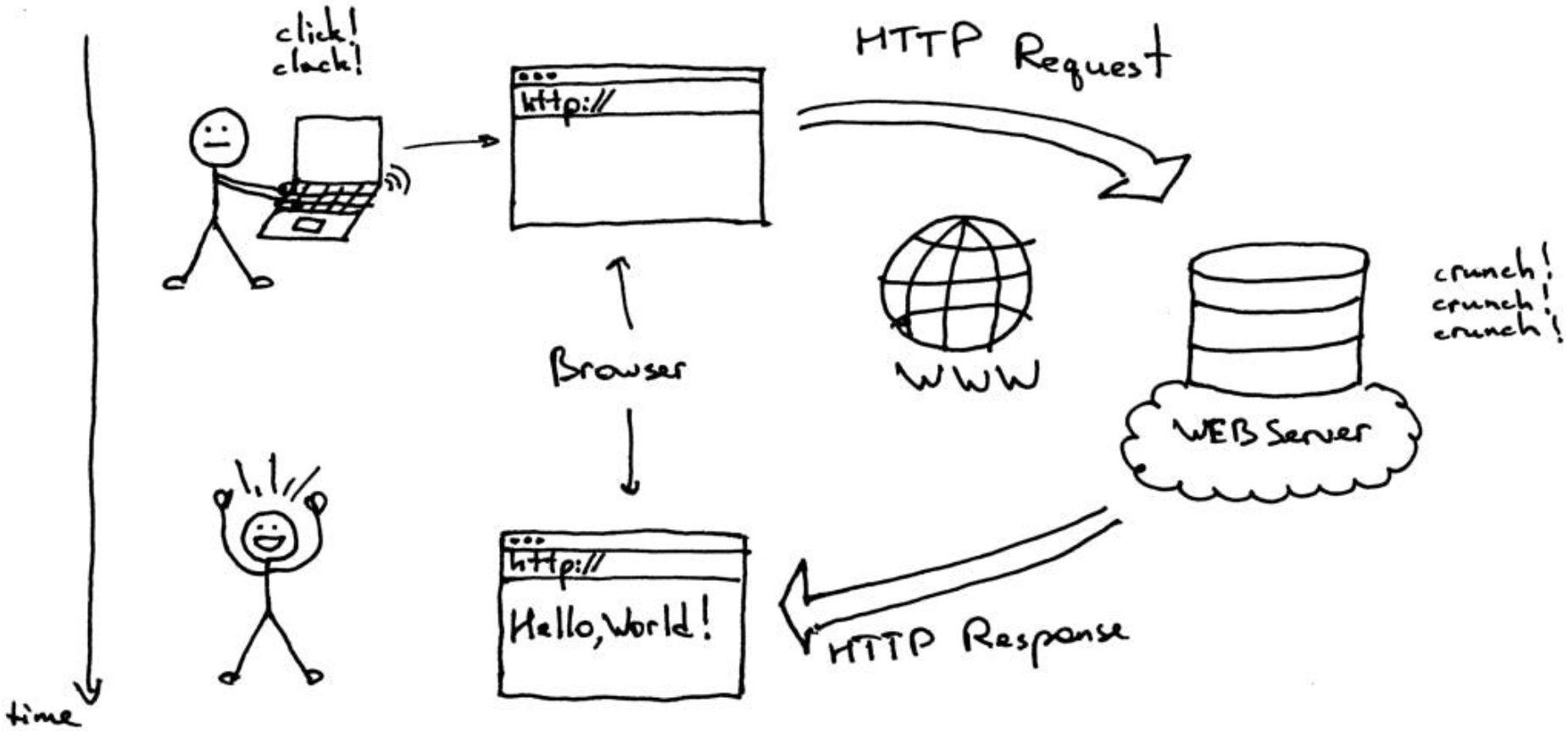
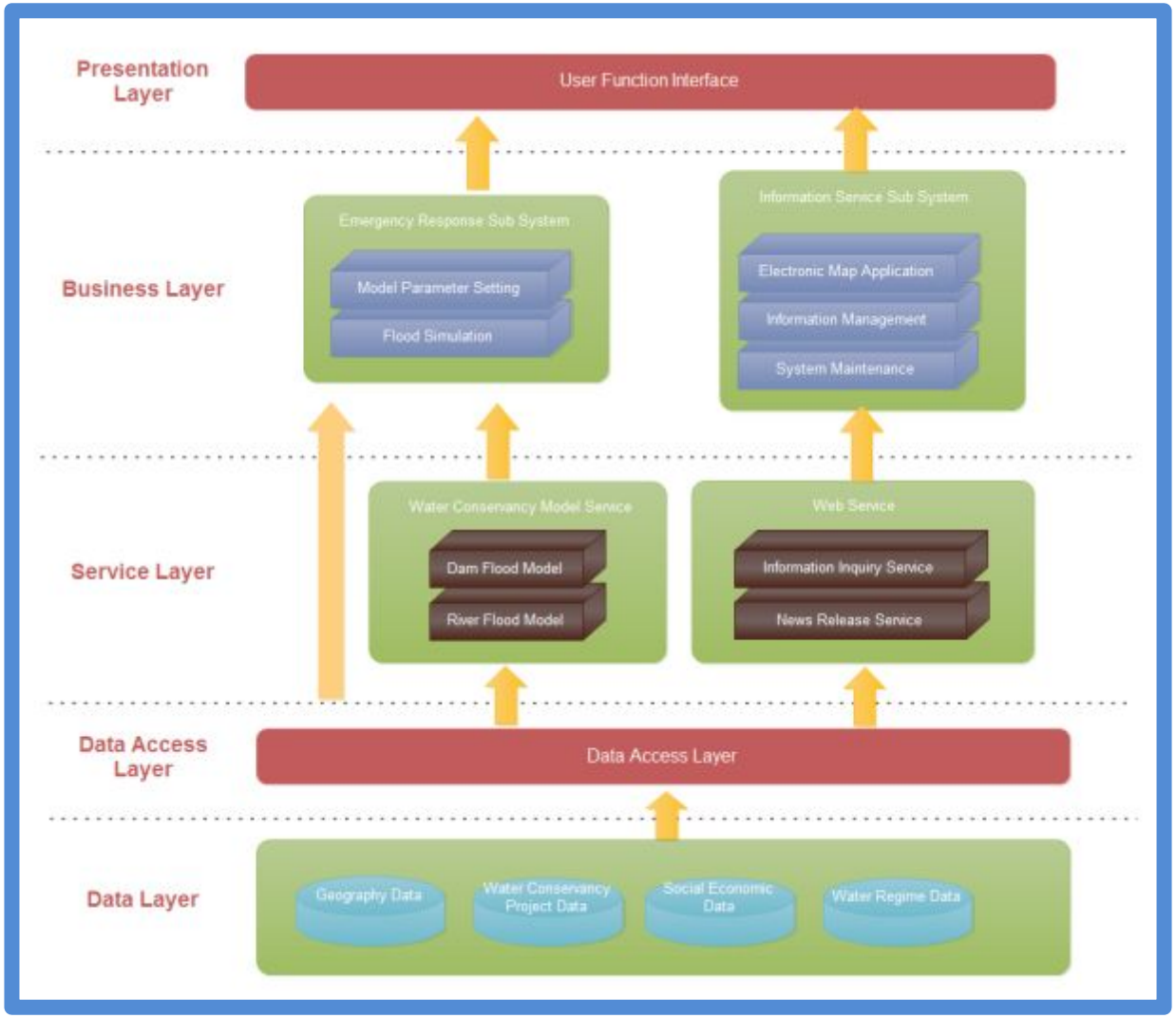
# Giới thiệu
# về Kiến trúc phần mềm

# Kiến trúc phần mềm



# Kiến trúc phần mềm



# Định nghĩa KTPM

- Có nhiều trường phái
- Richard N. Taylor và đồng nghiệp
  - A software system's architecture is **the set of principal design decisions** about the system
  - Kiến trúc phần mềm là bản thiết kế cho việc xây dựng và tiến hóa phần mềm
  - Các quyết định thiết kế bao gồm nhiều khía cạnh của hệ thống đang được phát triển

# Định nghĩa KTPM

L. Bass và đồng nghiệp: The software architecture of a system is the set of **structures** needed to reason about the system, which comprise **software elements**, **relations** among them, and **properties** of both

# Định nghĩa KTPM

IEEE: The **fundamental organization of a system**, embodied in its **components**, their **relationships to each other** and the **environment**, and the **principles** governing its design and evolution

# Kiến trúc phần mềm

- KTPM nhằm đáp ứng các yêu cầu chức năng và phi chức năng
- KTPM là một sự trừu tượng hóa
- KTPM xác định cấu trúc
- KTPM xác định tương tác giữa các thành phần

# Tại sao KTPM quan trọng?

# Tại sao KTPM quan trọng?

- Hạn chế/hỗ trợ cho thuộc tính của hệ thống
  - Có mối liên hệ chặt chẽ giữa kiến trúc và thuộc tính chất lượng
  - Ví dụ: phương thức giao tiếp giữa các thành phần phụ thuộc vào tính chất của tính năng. Ứng dụng chat?

# Tại sao KTPM quan trọng?

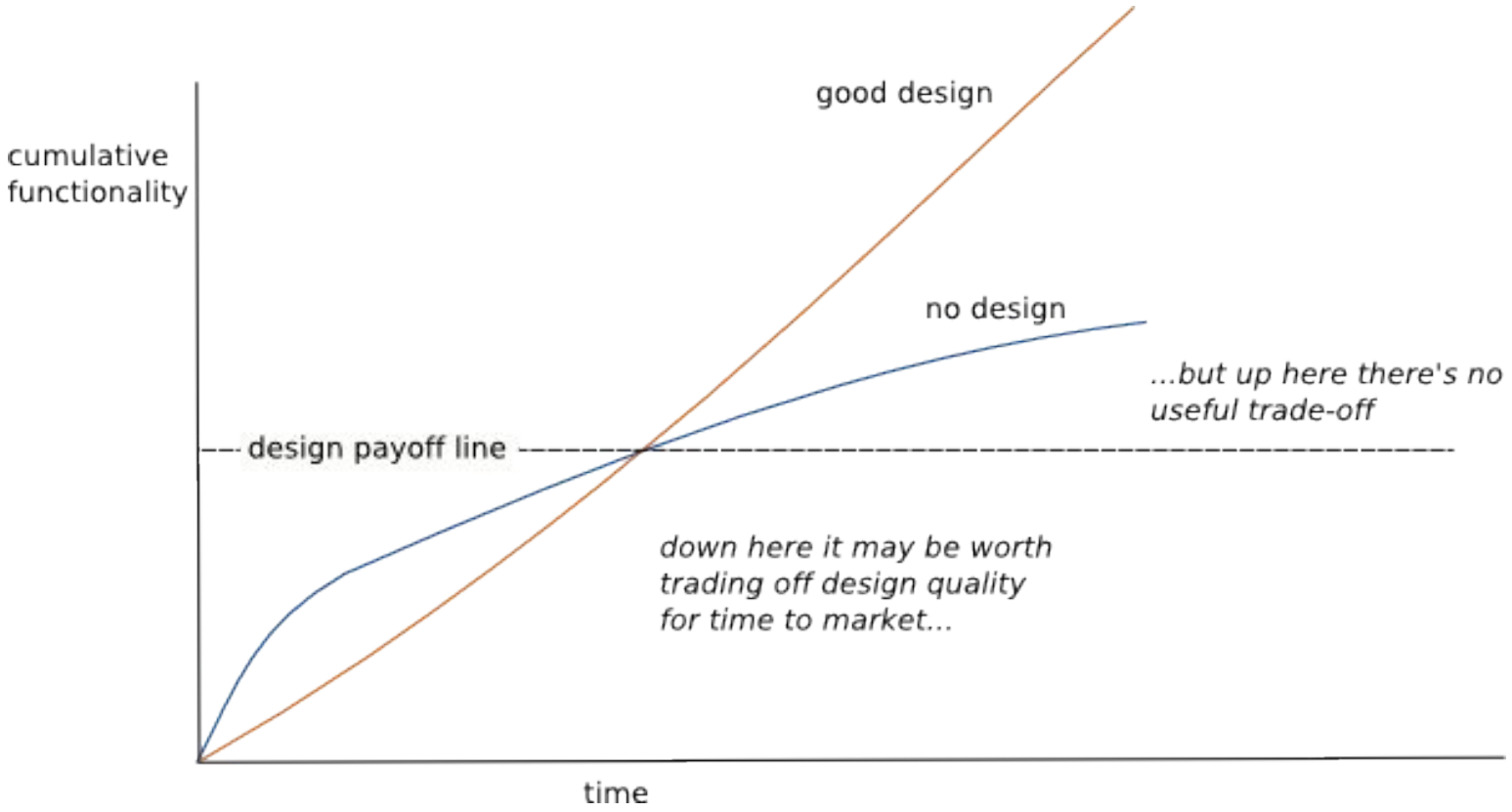
- Lập luận và quản lý thay đổi
  - Thay đổi gần như là điều tất yếu trong việc xây dựng hệ thống phần mềm
  - Mong muốn: thay đổi chỉ thực hiện trong các phần tử (method, class, package)
  - Kiến trúc hỗ trợ xác định sự thay đổi cần thiết không, nên thay đổi như thế nào, hậu quả/kết quả của việc thay đổi
  - Phân tích kiến trúc để tránh “architecture debt”

# Tại sao KTPM quan trọng?

- Phương tiện giao tiếp giữa các bên liên quan
  - Kiến trúc là một sự trừu tượng hóa
  - Dễ dàng cho các bên liên quan có thể hiểu và trao đổi các vấn đề liên quan đến hệ thống (đã, đang, hoặc sẽ xây dựng)
  - Các bên liên quan: người dùng, khách hàng, quản lý dự án,...

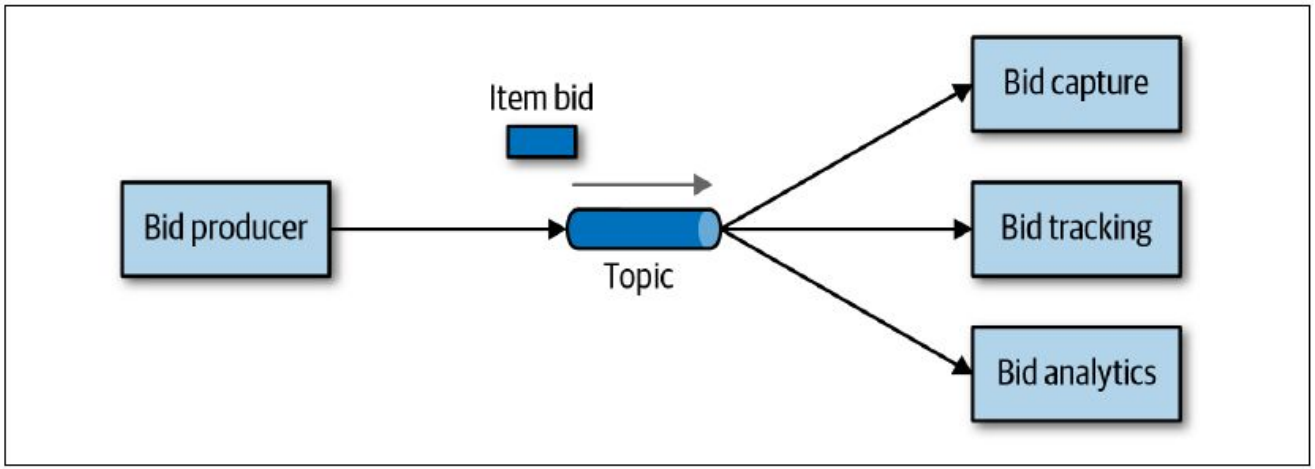
# Tại sao KTPM quan trọng?

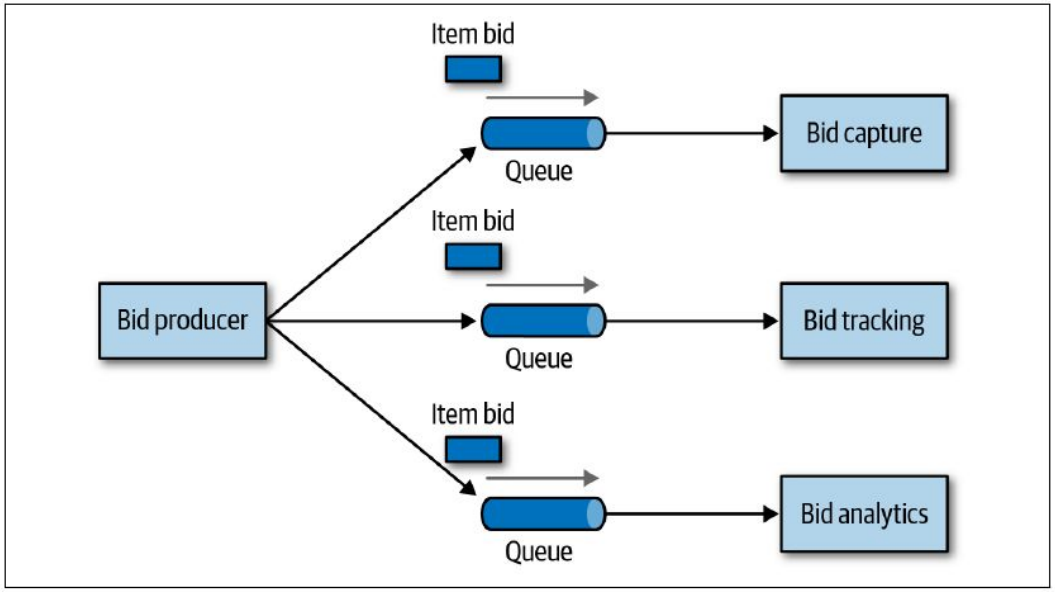
- Đặt ra các ràng buộc cho việc cài đặt
- Ảnh hưởng đến cấu trúc tổ chức của nhóm/bộ phận/doanh nghiệp xây dựng phần mềm
- Hỗ trợ quy trình phát triển mở rộng ngang/sâu
- Hỗ trợ xác định chi phí và kế hoạch thực hiện
- Chuyển giao, tái sử dụng


good design
cumulative
functionality
no design
...but up here there's no
useful trade-off
design payoff line
down here it may be worth
trading off design quality
for time to market...
time

# Kiến trúc sư phần mềm

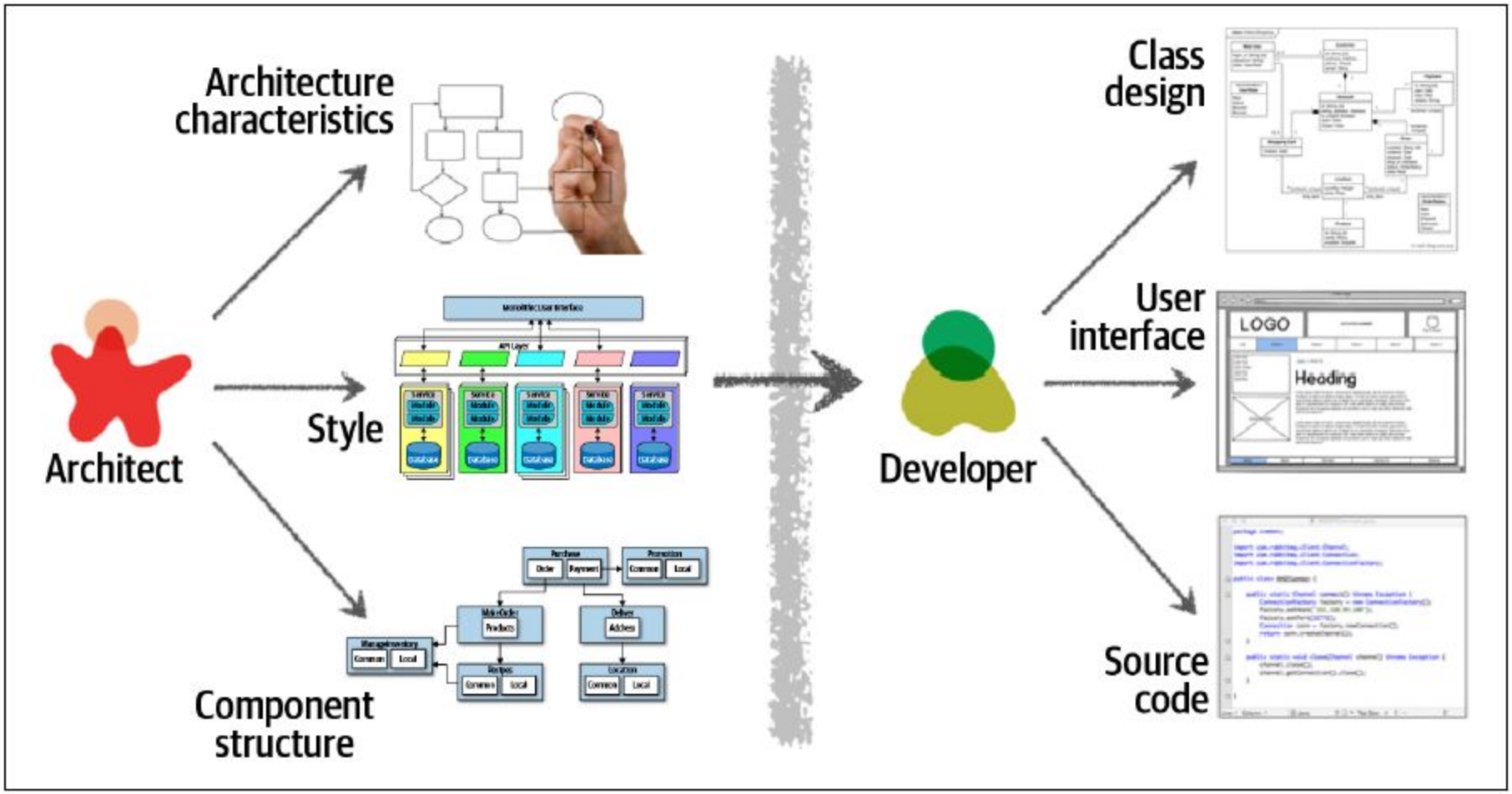
- Vai trò
  - Nắm được các yếu tố quyết định kiến trúc (architectural drivers)
  - Thiết kế kiến trúc
  - Biết được các rủi ro kỹ thuật
  - Lãnh đạo về kỹ thuật
  - Đảm bảo chất lượng

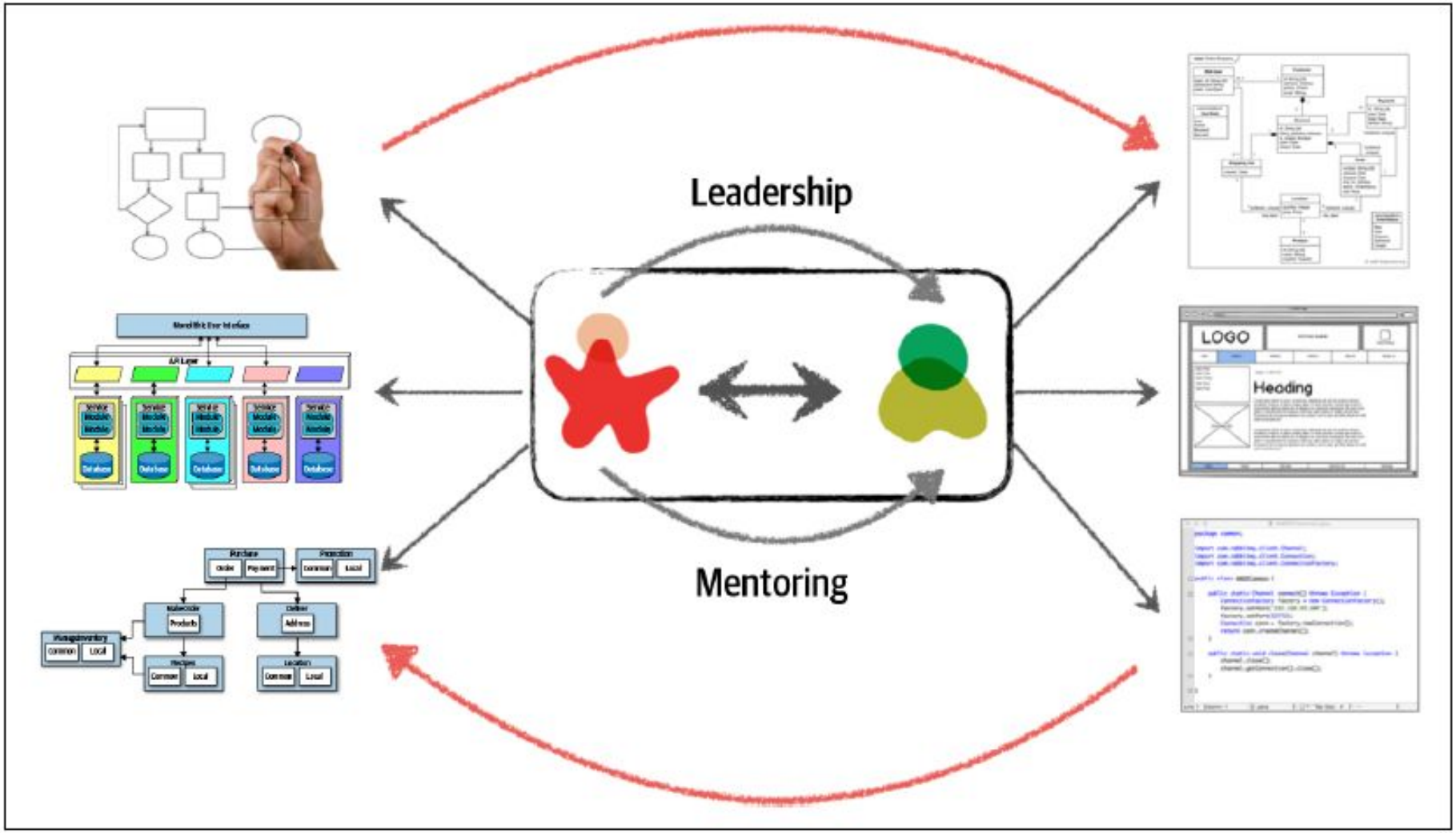

Item bid
Bid producer
Topic
Bid capture
Bid tracking
Bid analytics


Item bid
Queue
Bid capture
Item bid
Bid producer
Queue
Bid tracking
Item bid
Queue
Bid analytics

# Kỹ năng và kiến thức

- Kỹ thuật
- Kỹ năng mềm
- Kiến thức miền


Architecture characteristics
Architect
Style
Component structure
Developer
Class design
User interface
Source code
LOGO
Heading


Leadership
Mentoring
LOGO
Heading

# 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

# 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

# Thuộc tính chất lượng

- Tại một thời điểm, hệ thống có thể phục vụ bao nhiêu người?
- Khi 1000 người dùng truy cập, người dùng mất bao lâu để chuyển trang?

Hiệu năng

- Hệ thống được thiết kế để phục vụ 10K người dùng đồng thời. Khi con số này là 15K
  - Hệ thống có tự động tăng khả năng phục vụ không?
  - Có thể cắm thêm RAM, CPU,.. không? Cài đặt thêm vào máy chủ khác không?

Khả năng mở rộng

- Trong 1 tháng, hệ thống bị "down" mấy lần (vì nâng cấp, bảo trì, OutOfMemory, hỏng phần cứng,...)
- Khoảng thời gian bị "down" là bao nhiêu?

Tính sẵn sàng (Availability)

- Người dùng có cần đăng nhập để sử dụng hệ thống không?
- Hệ thống có được bảo vệ bằng nhiều tầng không?
- Dữ liệu của người dùng có được mã hóa không?

Bảo mật

- Hệ thống có thể sử dụng phương thức thanh toán qua thẻ tín dụng, ATM, Momo,...không?
- Hệ thống có thể kết nối với Grab Food, Foody,... không?

Khả năng tích hợp

- Giải thuật tìm kiếm theo địa điểm chưa hiệu quả, nâng cấp được không?
- Khi nâng cấp giải thuật tìm kiếm thì số lượng mô đun bị ảnh hưởng có nhiều không?
- Bổ sung tính năng gợi ý món ăn phù hợp được không?

Khả năng thay đổi

- Cách bố trí menu, thông báo,...có thuận tiện cho người dùng không?
- Khi người dùng gõ từ khóa vào ô tìm kiếm, hệ thống có hỗ trợ “auto complete” không?
- Người dùng có thể “Cancel” hay đổi món sau khi đặt không?
- Dùng hệ thống qua thiết bị di động được không?

Tính dễ sử dụng

(Usability)

# Kiến trúc và phát triển linh hoạt

- Có cần thiết kế kiến trúc khi áp dụng phương pháp phát triển linh hoạt không?