



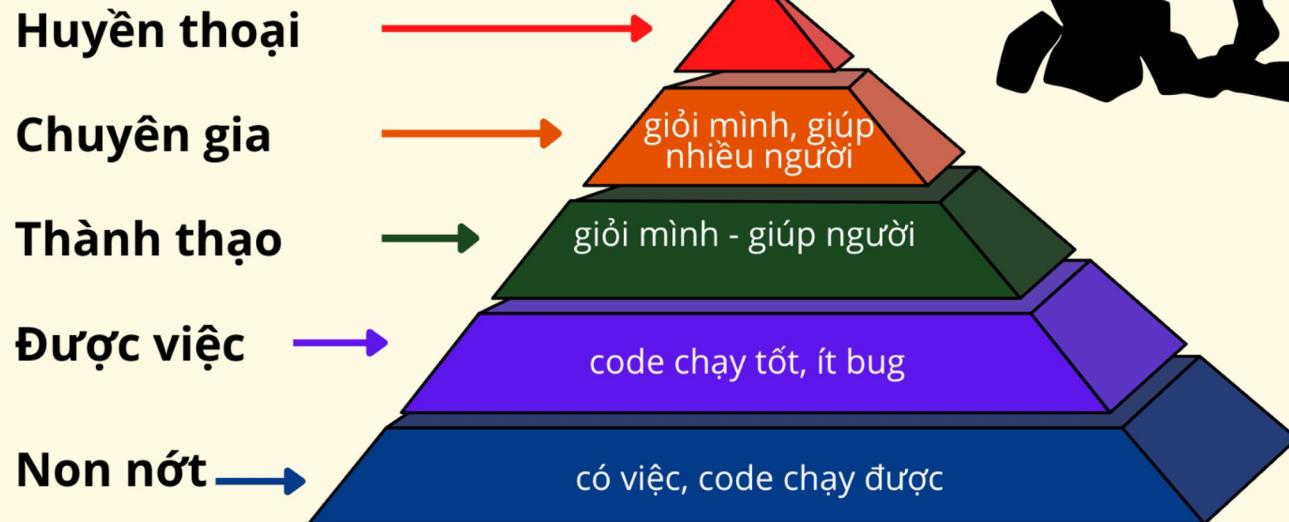
Workshop 1: Đóng gói

Trao đổi, thảo luận về việc đóng gói phần mềm & 1 vài tips mở rộng mạng lưới quan hệ cho DEV

HÔM NAY TA NÓI VỀ CÁI GÌ

- 1 vài khó khăn, thắc mắc, điểm lại các góp ý
- Đi sâu hơn về đóng gói - những nguyên lý và giải thích sâu
- Chém gió 1 chút về những cách để mở rộng mạng lưới network cho anh em DEV

Tháp DEV



Trước khi trở thành người hùng,
bạn phải sống sót đã

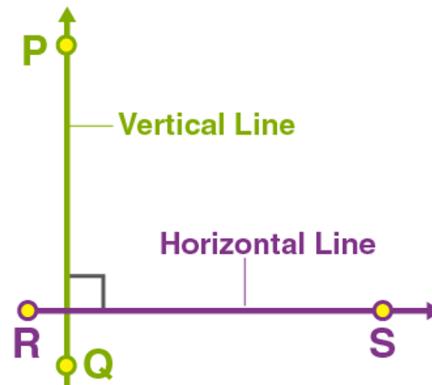


Những thế lưỡng nan của DEV

- Đi sâu hay đi rộng?
- Code sạch hay code nhanh?
- Đam mê hay tiền?

Đi sâu hay đi rộng?

- Em muốn làm fullstack
- Em muốn sống với product mỗi ngày



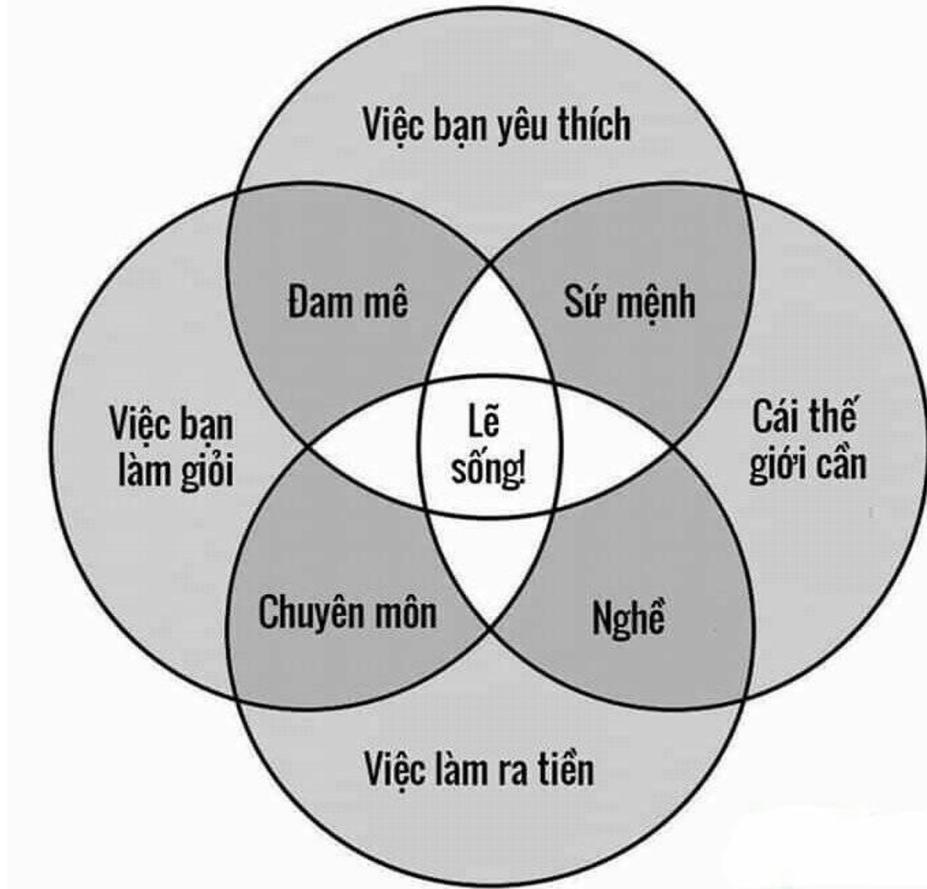
Clean or Speed?

- Clean Code: SOLID, DRY, KISS, YAGNI
- Timeline, Deadline, KPI, OKR



Đam mê hay tiền?

- Tiền
- Mê
- Thích



Những thế lưỡng nan của tổ chức / quản lý

- Ngắn hạn (performance) hay dài hạn (maintenance)?
- Thuyết X hay thuyết Y?
- Tái cấu trúc (refactoring) hay đập đi xây lại (rewrite)?

Ngắn hạn hay dài hạn?

Giả định, khi cần tuyển dụng 1 DEV vào team cho sản phẩm hiện tại và sắp tới, bạn có bao giờ phân vân giữa:

- 1 dev khá, đã làm qua sản phẩm tương tự ở mức độ quen tay, có thái độ bình thường
- 1 dev có bằng giỏi đại học, chưa có nhiều sản phẩm hoặc ít kinh nghiệm thực tế, thái độ cầu thị



X or Y?

- **Thuyết X** cho rằng nhân viên không có động lực nội tại để làm việc, lẩn tránh trách nhiệm, hướng tới lợi ích cá nhân
- **Thuyết Y** cho rằng nhân viên có động lực nội tại để làm việc, nhận trách nhiệm, hướng tới hoàn thiện bản thân và tổ chức

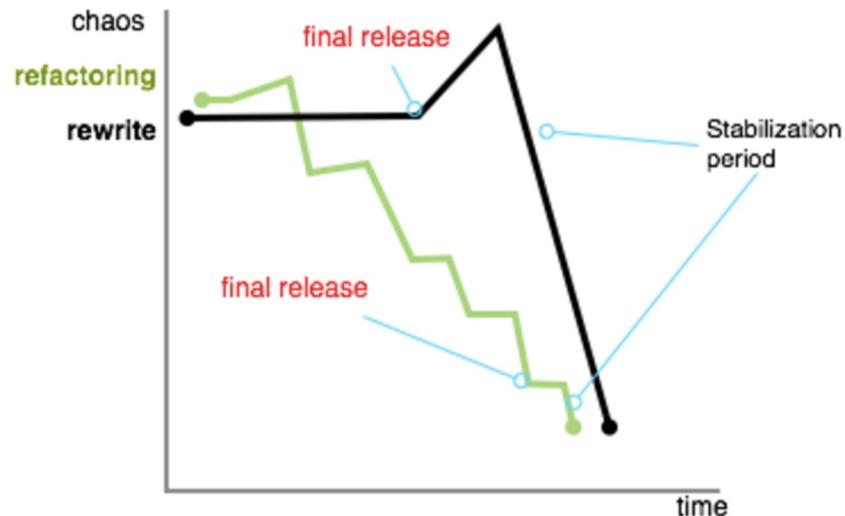


THEORY X AND THEORY Y

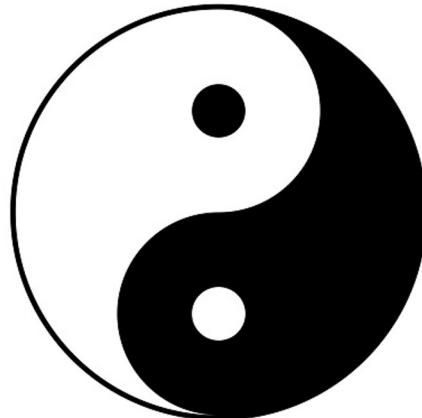
DOUGLAS McGREGOR (1906-1964)

Refactoring or Rewrite?

- Code của thằng khác luôn thối hơn mình
- Dev đến và đi, code ở lại



Giống như bit chỉ có 0 và 1, DEV thích
đúng sai rõ ràng, nhưng cuộc sống
có phải chỉ có đen và trắng?



Think Big, Start Small, Move Fast



Nguyên lý

- Project approach
- Test first
- ROI driven
- Data driven
- Systemic thinking
- Triangle feedbacks



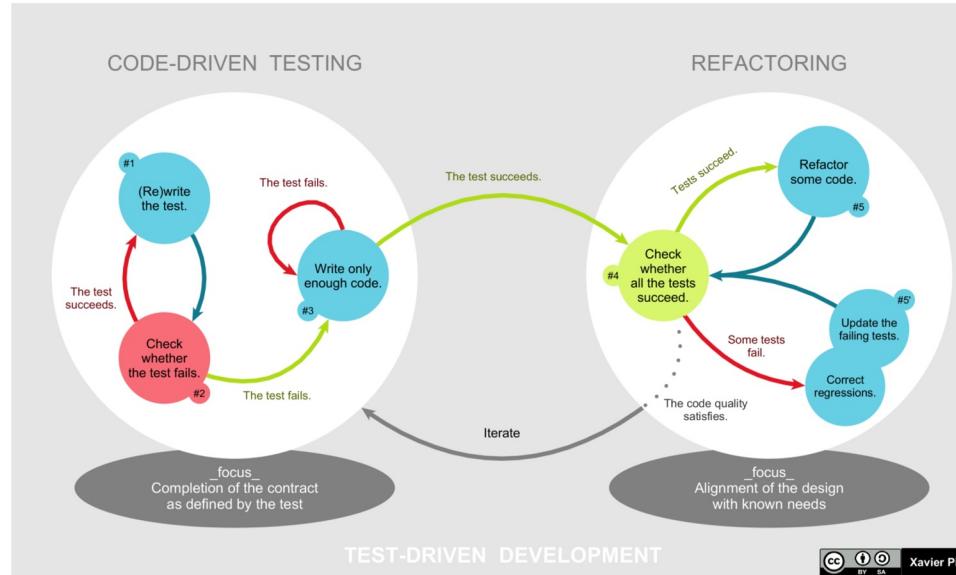
Project Approach

1. Định nghĩa rõ: mục đích và kết quả
2. Lập phương án và chiến lược khả thi
3. Lập và quản lý kế hoạch
4. Thực thi và quản lý quá trình thực thi
5. Đánh giá mức độ đạt được của mục đích và kết quả



Test first

- Biết rõ phạm vi công việc và kết quả mong muốn
- Biết rõ lịch trình dự kiến triển khai
- Có đánh giá sau khi hoàn thành



ROI driven

- Biết rõ có gì kém và tốt hơn
- Ưu nhược điểm dưới góc nhìn của tổ chức
- ROI với DEV liên quan gì



Data driven

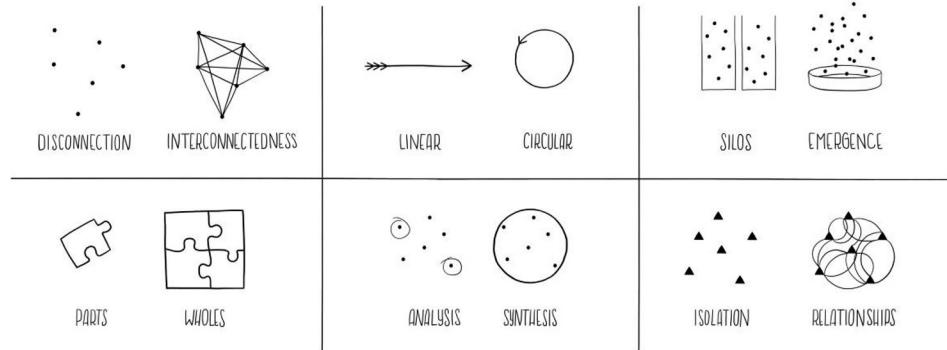
- Lượng hoá mọi yếu tố có thể
- Cố gắng lượng hoá nhiều nhất có thể
- Các báo cáo công khai của các tổ chức độc lập: mức độ tăng trưởng của stack, kiến trúc
- Các báo cáo nội bộ của tổ chức: performance, crash, response time



Systemic thinking

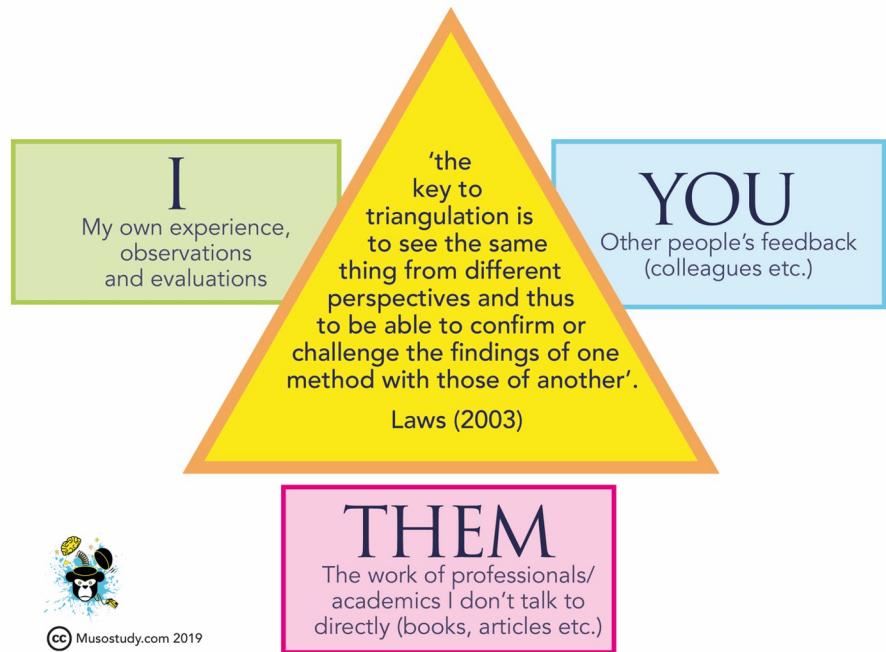
- Data driven và ROI driven chỉ là mindset, khi bắt tay vào làm cần sử dụng tư duy toàn cảnh
- Nâng cao tính minh bạch và cộng tác giữa các team member
- Brainstorm liên tục trong quá trình làm, cải tiến

TOOLS OF A SYSTEM THINKER



Triangle feedbacks

- 3 thành tố phản hồi: quản lý trực tiếp, đồng nghiệp, khách hàng (cộng đồng)
- Tiếp nhận phản hồi giúp chủ động tìm kiếm giải pháp và thay đổi cho phù hợp
- Không bao giờ nói “Không”, hãy nói “Có thể”



CC Musostudy.com 2019

Yếu tố

- OOP
- SDLC
- Packager
- Maintainer
- Collection



OOP

- Tính Đóng gói
- Tính Kế thừa
- Tính Trùu tượng
- Tính Đa hình

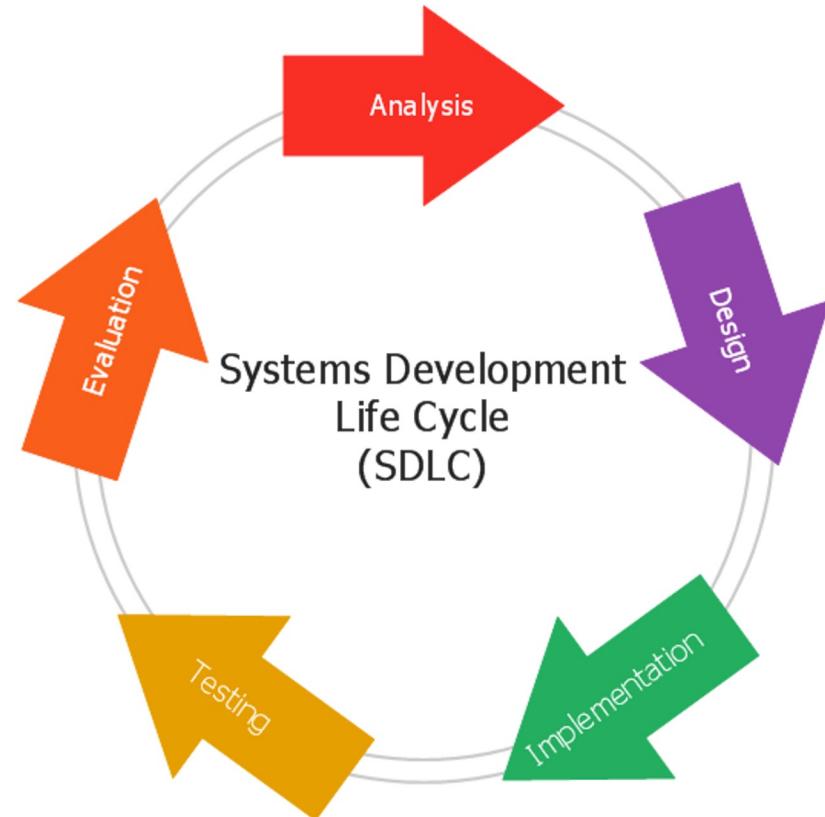
```
private static boolean depthFirstSearch(int source, int destination){  
    Node s = getNode(source);  
    Node d = getNode(destination);  
    HashSet<Integer> visited = new HashSet<>(); // Keeps track of ids already visited  
    return depthFirstSearch(s, d, visited);  
}  
  
private static Boolean depthFirstSearch(Node source, Node destination, HashSet<Integer> visited){  
    // Check if this node is already visited  
    if(visited.contains(source.id))  
        return false; // No path for this problem if node is already visited  
      
    // Mark node as visited  
    visited.add(source);  
      
    // Check if source id matches target  
    if(source == destination){  
        return true; // We found a path  
    }  
      
    // Otherwise make more subproblems. For each node adjacent to the subgraph of source.  
    for(Node n: source.adjacent){  
        if(depthFirstSearch(n, destination, visited)){  
            return true;  
        }  
    }  
      
    return false; // Otherwise no path  
}
```

.....
..... Breadth First Search

What Is OOP?

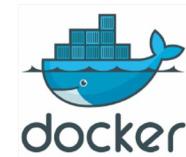
Life Cycle

- Phân tích tính năng rõ ràng
- Thiết kế Input / Output và Interface để giao tiếp ra bên ngoài rõ ràng
- Triển khai tính năng
- Kiểm thử theo kịch bản và tích hợp vào module sản phẩm
- Peer review, feedback về tính năng



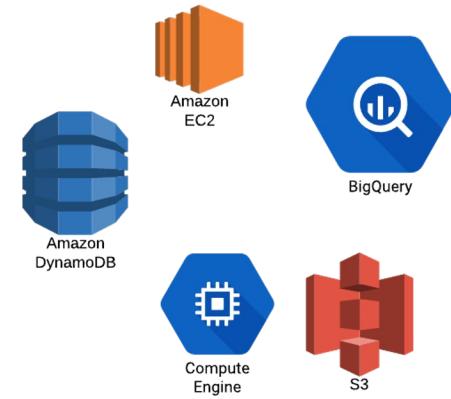
Packager

- Sử dụng Package Manager để đóng gói code
- Chia nhỏ các tính năng theo mô hình Micro services
- Tích hợp, phát triển và bảo trì liên tục
- Codebase luôn được nâng cấp, hoàn thiện, và thích hợp với nhiều version, nền tảng, hệ điều hành



Maven™

Packaged



Managed

Document

- Rõ ràng, mạch lạc, đơn giản, dễ hiểu
- Document viết cho người khác, không phải viết cho mình
- Document viết nên có cảm xúc, có mở vấn đề, cách giải quyết vấn đề và kết thúc vấn đề (mở bài - thân bài - kết luận)
- Phải được Update

BS-BaseAPI-Laravel

Base API Laravel



Bắt đầu

Bộ Base API Laravel này sử dụng package `beetsoftweb` từ repository `bs-base-api-lite` implement vào framework laravel để viết. Base API là một API RESTful dựa trên các request và trả về response dưới dạng JSON. Bộ document được viết ở postman kèm các api mẫu giúp người dùng dễ dàng kết nối và sử dụng.

Register user (IMPORTANT)

- * Khi đăng ký người dùng sẽ được cấp 1 `username (client_id)` và `secretkey`.
- * `client_id` và `secret_key` được dùng để mã hóa thông tin khi gửi request xác thực thông tin người gửi lên hệ thống.

Các biến và response trả về

1. Variables postman

Các biến dùng chung của các api thì được đặt trong variables của collection

Ví dụ:

2. Chi tiết một api

Ngoài các input truyền lên, 1 params của api sẽ cần truyền thêm `username` (tức `client_id` được cấp và `signature`).

Signature truyền lên sử dụng thuật toán md5 mã hóa chuỗi các value của từng api quy định, nối với nhau bằng ký tự `PREFIX_AUTH=$` với `client_id`, `secret_key`

Ví dụ một signature của api config được tạo như sau:

Api yêu cầu signature truyền lên là mã hóa chuỗi gồm `id`, `language`, `client_id`, `secret_key` nối với nhau bởi `PREFIX_AUTH` sau:

JUMP TO
Introduction
> Category
> Auth
> Config
> User
> Option
> Tag
> Topic
> DepartmentStructure
> Permission
> User Group
> Menu
> Link
> Post
> Banner
> Language
> ProbeOpinion
> Question
> Source

Maintainer

- Luôn có người phụ trách về package
- Liên tục đánh giá, cải tiến và báo cáo về tính năng
- Lượng hoá các báo cáo



Collection

- Liên tục thu thập, đánh giá và hoàn thiện kho thư viện của team
- Liên tục nâng cấp tính năng cho từng gói
- Các gói cũng nên có sự tiến hoá cho hợp thời cuộc





Những yếu tố làm lên giá trị

Triệt tiêu nỗi đau của những thế lưỡng nan
đã nói bên trên

Giúp người quản lý có những phương án dự
phòng rủi ro cho sự lựa chọn

Giúp dev tự do thoải mái hơn 1 chút với cá
tính của mình



ĐƯỜNG QUÊN, ĐIỀU QUAN TRỌNG

Mọi người có cơ hội sửa sai (cả dev và quản lý), cải tiến những đoạn code, những gói mình đã viết ra bất cứ lúc nào, kể cả là khi dự án đã kết thúc và không còn cần phải sửa / nâng cấp nữa.

Qua đó làm mới / tốt hơn mỗi ngày, từ đó mà càng về sau, yếu tố kinh nghiệm, gắn bó, rèn luyện kỹ năng của dev ngày càng được nâng cao.

Thảo luận: Cách để mở rộng mạng lưới mối quan hệ cho DEV

1 vài cách

- Giống như ly nước, đã đầy rồi thì làm sao mà rót thêm được nữa
- Áp dụng tháp nhu cầu Maslow để biết nên hỏi ai
- Open Source là 1 ý kiến không tồi để mở rộng mối quan hệ
- Tôn sư trọng đạo
- Bất cứ ai mình gặp trong đời, đều là duyên cả
- Mây tầng nào gặp mây tầng đó, đừng cố quá
- Ra ngoài nhiều hơn chỉ ngồi trước máy tính
- Mở lòng, chia sẻ câu chuyện của mình để giảm bớt sự đề phòng từ người khác



Workshop sau có gì?

- Trao đổi, thảo luận về kinh nghiệm vận hành, phát triển, theo dõi hệ thống, dịch vụ dành cho DEV
- Chia sẻ và thảo luận về SEO dành cho DEV, ở góc nhìn và vai trò DEV



Thanks & QA
Hẹn gặp các
chiến hữu ở
workshop sau