

DevOps Culture and Practice with OpenShift

Deliver continuous business value through people,
processes, and technology

Section 1: Practices Make Perfect



Văn hóa và thực hành DevOps với OpenShift

Mang lại giá trị kinh doanh liên tục thông qua con người, quy trình và công nghệ

Tim Beattie

Mike Hepburn

Noel O'Connor

Donal mùa xuân

Packt>

BIRMINGHAM—MUMBAI

Chào mừng bạn đến với Văn hóa và Thực hành DevOps với OpenShift.

Cuốn sách này sẽ cho phép người đọc tìm hiểu, hiểu và áp dụng nhiều phương pháp thực hành khác nhau - một số liên quan đến con người, một số liên quan đến quy trình, một số liên quan đến công nghệ - để giúp việc áp dụng DevOps và từ đó OpenShift thành công trong tổ chức của họ. Nó giới thiệu nhiều khái niệm và công cụ DevOps mà chúng tôi sử dụng để kết nối văn hóa và thực tiễn DevOps thông qua một vòng lặp liên tục khám phá, xoay trục và phân phối. Tất cả những điều này được củng cố bởi nền tảng văn hóa, hợp tác và kỹ thuật.

Cuốn sách này cung cấp một tập bản đồ trình bày cách xây dựng các nhóm sản phẩm được trao quyền trong tổ chức của bạn. Thông qua sự kết hợp của các câu chuyện trong thế giới thực, một trường hợp sử dụng bìa đặt (đặc biệt thú vị đối với những người yêu chó và mèo), hướng dẫn hỗ trợ và chi tiết kỹ thuật về cách triển khai tất cả, cuốn sách này cung cấp các công cụ và kỹ thuật để xây dựng văn hóa DevOps trong tổ chức của bạn trên Nền tảng vùng chứa OpenShift của Red Hat.

Đó là một bộ sưu tập các sách về tư duy thiết kế, DevOps, văn hóa, hỗ trợ và hỗ trợ kỹ thuật thực hành tất cả trong một! Như Gabrielle Benefield (nhà lãnh đạo tư tưởng về Kết quả kinh doanh) giải thích trong lời nói đầu, cuốn sách này "giống như có một cuốn cảm nang du lịch tuyệt vời mà bạn có thể rút ra trong hành trình của mình, nó mang đến cho bạn phương hướng và ý tưởng bạn cần khi bạn cần. Tôi sử dụng cuốn sách này cuốn sách này như một tài liệu tham khảo đáng tham khảo mà tôi có thể cung cấp cho các đội để giúp họ đứng dậy và chạy nhanh. Tôi cũng thích việc các tác giả nói chuyện một cách thẳng thắn và chia sẻ những câu chuyện chiến tranh trong thế giới thực của họ, bao gồm cả những sai lầm và chàm bầy."

Để giúp bạn hiểu rõ hơn về cuốn sách, 18 chương đã được sắp xếp thành 7 phần:

➤ Phần 1, Thực hành tạo nên sự hoàn hảo giới thiệu về văn hóa và thực tiễn DevOps. Nó cũng sẽ cung cấp cái nhìn tổng quan về công cụ điều hướng mà chúng tôi sẽ sử dụng để tìm hiểu cách chúng tôi sử dụng khả năng khám phá liên tục và phân phối liên tục để đạt được văn hóa DevOps.

- Phần 2, Thiết lập nền tảng cung cấp các phương pháp thực hành mà chúng tôi sử dụng để thiết lập một nền văn hóa mở cho phép các nhóm có hiệu suất cao hiện thực hóa DevOps và các phương pháp thực hành nền tảng kỹ thuật mà họ sử dụng để khởi động nhằm đạt được DevOps.
- Phần 3, Discover It giải thích các phương pháp mà chúng tôi sử dụng để khám phá lý do, cho ai và cách chúng tôi xây dựng các sản phẩm ứng dụng tuyệt vời để chạy trên OpenShift và mang lại giá trị kinh doanh sớm và liên tục.
- Phần 4, Ưu tiên Nó cho thấy cách chúng tôi quyết định những việc cần làm bằng cách áp dụng phương pháp thử nghiệm và cách chúng tôi tổ chức công việc của mình theo giá trị kinh doanh và rủi ro.

- Phần 5, Phân phối bao gồm các phương pháp phân phối Agile và thác nước cũng như các kỹ thuật mà chúng tôi sử dụng để đo lường và học hỏi, ở nhiều cấp độ, từ phân phối lặp lại và phân phối tăng dần.
- Phần 6, Xây dựng, vận hành, sở hữu Nó xem xét công nghệ và hướng dẫn các bước, mẫu và công cụ mà chúng tôi sử dụng để tự tin cung cấp và vận hành ứng dụng và nền tảng nghiên cứu điển hình của mình.
- Phần 7, Cải thiện nó, Duy trì nó mô tả cách chúng tôi tiếp tục vòng lặp vô hạn để liên tục tìm hiểu và cải tiến các sản phẩm cũng như công nghệ của mình cũng như cách áp dụng mô hình tư duy tương tự được sử dụng cho các sản phẩm ứng dụng cho các nền tảng và chiến lược.

Mục lục

Lời nói đầu	iii
<hr/>	
Phần 1: Thực hành để hoàn thiện	1
<hr/>	
Chương 1: Giới thiệu - Bắt đầu với Tại sao	3
<hr/>	
Tại sao - Vì lý do hoặc mục đích gì?	4
Tại sao tôi nên lắng nghe những người này?	5
Cuốn sách này đến từ đâu?	6
Chính xác thì cuốn sách này dành cho ai?	8
Từ I đến T đến M.....	10
Phần kết luận	11
<hr/>	
Chương 2: Giới thiệu DevOps và một số công cụ	13
<hr/>	
Chuỗi giá trị.....	14
Khoảng trống	16
Danh sách lớn những việc cần làm.....	16
Thể hiện giá trị và xây dựng điều đúng đắn	17
Chúng ta thực hiện những việc trong danh sách của mình như thế nào?	18
Phát triển đến vận hành.....	22
Con người, quy trình và công nghệ.....	24
Vòng lặp Mobius và Thư viện Thực hành Mở	26
Kết luận	32

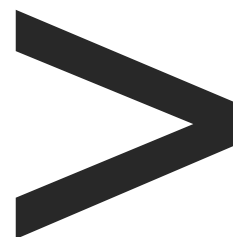
Chương 3: Hành trình phía trước 33

Câu chuyện về việc kể một bài tập.....	35
PetBattle – Cốt truyện	36
Còn các Hệ thống kế thừa thì sao?	37
Vay mượn sự sáng chói	38
Điều gì sẽ xảy ra ở phần còn lại của cuốn sách này?	38

Còn các Nhóm phân phối thì sao? 43

Một số lời nói về thế giới 'Mở'..... 44

Phân kết luận 45



Lời nói đầu

Về

Phần này giới thiệu ngắn gọn về các tác giả, nội dung của cuốn sách này, những kỹ năng bạn cần để bắt đầu cũng như phần cứng và phần mềm cần thiết để hoàn thành tất cả các chủ đề kỹ thuật.

Giới thiệu về Văn hóa DevOps và thực hành với OpenShift

Văn hóa và thực hành DevOps với OpenShift có nhiều phương pháp thực hành trong thế giới thực khác nhau - một số liên quan đến con người, một số liên quan đến quy trình, một số liên quan đến công nghệ - để tạo điều kiện cho DevOps thành công và đến lượt OpenShift, việc áp dụng trong tổ chức của bạn. Nó giới thiệu nhiều khái niệm và công cụ DevOps để kết nối văn hóa và thực hành thông qua một vòng lặp liên tục khám phá, chuyên hướng và phân phối được củng cố bởi nền tảng cộng tác và công nghệ phần mềm.

Các bộ chứa và quản lý vòng đời ứng dụng tập trung vào bộ chứa hiện là một tiêu chuẩn ngành và OpenShift có vị trí dẫn đầu trong một thị trường đang phát triển mạnh mẽ về các dịch vụ sản phẩm dựa trên Kubernetes dành cho doanh nghiệp. Văn hóa và thực hành DevOps với OpenShift cung cấp lộ trình xây dựng các nhóm sản phẩm được trao quyền trong tổ chức của bạn.

Hướng dẫn này tập hợp tất cả các tính năng tinh gọn, linh hoạt, tư duy thiết kế, DevOps, văn hóa, hỗ trợ và hỗ trợ kỹ thuật thực hành trong một cuốn sách. Thông qua sự kết hợp giữa các câu chuyện trong thế giới thực, nghiên cứu trường hợp thực tế, hướng dẫn hỗ trợ và chi tiết triển khai kỹ thuật, Văn hóa và thực hành DevOps với OpenShift cung cấp các công cụ và kỹ thuật để xây dựng văn hóa DevOps trong tổ chức của bạn trên Nền tảng vùng chứa OpenShift của Red Hat.

Về các tác giả

Tim Beattie là Giám đốc Sản phẩm Toàn cầu và Trưởng nhóm Tương tác Hiệu trưởng Cấp cao của Phòng thí nghiệm Đổi mới Mở Red Hat. Sự nghiệp phân phối sản phẩm của ông kéo dài 20 năm với tư cách là huấn luyện viên chuyển đổi linh hoạt và tinh gọn - một người ủng hộ tư duy thiết kế và phân phối liên tục, người gắn kết mọi người lại với nhau để xây dựng các sản phẩm và dịch vụ có ý nghĩa đồng thời chuyển đổi các tập đoàn lớn hơn sang hướng kinh doanh linh hoạt. Anh ấy sống ở Winchester, Vương quốc Anh, cùng với vợ và con chó của mình, Gerrard the Labrador (Phòng thí nghiệm khác trong cuộc đời anh ấy) đã thích nghi từ việc trở thành người mèo thành người chó ở độ tuổi 30.

Mike Hepburn là Kiến trúc sư chính toàn cầu của Phòng thí nghiệm đổi mới mở Red Hat và giúp khách hàng thay đổi cách làm việc của họ. Anh dành phần lớn thời gian trong ngày làm việc của mình để giúp khách hàng và nhóm chuyên đổi cách họ chuyển ứng dụng sang sản xuất bằng OpenShift. Anh là đồng tác giả cuốn sách "DevOps with OpenShift" và yêu thích hoạt động ngoài trời, gia đình, bạn bè, cà phê ngon và bia ngon. Mike yêu hầu hết các loài động vật, không phải những con nhện lông to (Thợ săn) được tìm thấy ở Úc, và nói chung là một người yêu mèo trừ khi đó là Thứ Ba, khi anh ấy là một người yêu chó.

Noel O'Connor là Kiến trúc sư chính cấp cao trong Phòng Thực hành Giải pháp EMEA của Red Hat, chuyên về kiến trúc tích hợp và ứng dụng gốc trên nền tảng đám mây. Anh đã làm việc với nhiều khách hàng doanh nghiệp toàn cầu của Red Hat ở cả Châu Âu, Trung Đông và Châu Á. Anh ấy là đồng tác giả của cuốn sách "DevOps with OpenShift" và anh ấy không ngừng cố gắng học hỏi những điều mới để đạt được mức độ thành công khác nhau. Noel thích chó hơn mèo nhưng lại bị các thành viên còn lại trong nhóm phản đối.

Donal Spring là Kiến trúc sư cấp cao của Phòng thí nghiệm đổi mới mở Red Hat. Anh ấy làm việc trong các nhóm giao hàng với ông tay áo xắn lên để giải quyết mọi việc cần thiết - từ huấn luyện và cố vấn cho các thành viên trong nhóm, đưa ra định hướng kỹ thuật cho đến viết mã và viết bài kiểm tra. Anh ấy yêu thích công nghệ và bắt tay vào khám phá công nghệ, khuôn khổ và mô hình mới. Người ta thường thấy anh ấy vào cuối tuần đang viết mã cho các dự án cá nhân và tự động hóa tất cả mọi thứ. Mèo hay chó? Anh ấy thích cả hai :)

Về họa sĩ minh họa

Ilaria Doria là Trưởng nhóm Tương tác và Hiệu trưởng tại Phòng thí nghiệm Đổi mới Mở của Red Hat. Năm 2013, cô tham gia vào lĩnh vực Agile với vai trò là huấn luyện viên và hỗ trợ những khách hàng lớn trong hành trình chuyển đổi kỹ thuật số của họ. Nền tảng của cô là về trải nghiệm và tư vấn cho người dùng cuối bằng cách sử dụng các phương pháp mở để dẫn dắt quá trình chuyển đổi phức tạp và mở rộng quy mô linh hoạt trong các tổ chức lớn. Những tờ ghi chú và những bức vẽ nguệch ngoạc đầy màu sắc luôn là một phần trong cuộc sống của cô và đây là lý do tại sao cô cung cấp tất cả các hình ảnh minh họa trong cuốn sách và xây dựng tất cả các mẫu kỹ thuật số. Cô ấy chắc chắn là một người yêu chó.

Về người đánh giá

Ben Silverman hiện là Kiến trúc sư trưởng của nhóm Tài khoản Toàn cầu tại Cincinnati Bell Technology Services. Ông cũng là đồng tác giả của các cuốn sách OpenStack dành cho kiến trúc sư, Làm chủ OpenStack, OpenStack – Thiết kế và triển khai cơ sở hạ tầng đám mây, đồng thời là Người đánh giá kỹ thuật cho việc học OpenStack (Xuất bản Packt).

Khi Ben không viết sách, anh ấy hoạt động tích cực trong Ban Biên tập Siêu người dùng Cơ sở hạ tầng Mở và là người đóng góp kỹ thuật cho Nhóm Tài liệu Nền tảng Cơ sở hạ tầng Mở (Hướng dẫn Kiến trúc). Ông cũng lãnh đạo Nhóm người dùng cơ sở hạ tầng mở Phoenix, Arizona. Ben thường được mời phát biểu về việc áp dụng, triển khai, di chuyển và tác động văn hóa của đám mây và Kubernetes tại các sự kiện khách hàng, cuộc gặp gỡ và các phiên họp đặc biệt của nhà cung cấp.

Mục tiêu học tập

- Triển khai các phương pháp thực hành DevOps thành công và từ đó là OpenShift trong tổ chức của bạn
- Giải quyết vấn đề phân chia nhiệm vụ trong thế giới giao hàng liên tục
- Hiểu tự động hóa và tầm quan trọng của nó thông qua chế độ xem lấy ứng dụng làm trung tâm
- Quản lý các chiến lược triển khai liên tục, chẳng hạn như A/B, luân phiên, canary và blue-green
- Tận dụng khả năng Jenkins của OpenShift để thực hiện các quy trình tích hợp liên tục
- Quản lý và tách biệt cấu hình khỏi phần mềm thời gian chạy tĩnh
- Làm chủ giao tiếp và cộng tác cho phép phân phối các sản phẩm phần mềm ưu việt trên quy mô lớn thông qua việc khám phá liên tục và phân phối liên tục

Khán giả

Cuốn sách này dành cho bất kỳ ai quan tâm đến thực tiễn DevOps với OpenShift hoặc các nền tảng Kubernetes khác.

Cuốn sách DevOps này cung cấp cho các kiến trúc sư, nhà phát triển phần mềm và kỹ sư cơ sở hạ tầng hiểu biết thực tế về OpenShift, cách sử dụng nó một cách hiệu quả để triển khai hiệu quả các kiến trúc ứng dụng cũng như cách cộng tác với người dùng và các bên liên quan để mang lại kết quả có tác động đến doanh nghiệp.

Tiếp cận

Cuốn sách này kết hợp những giải thích lý thuyết chính xác với các ví dụ thực tế để giúp bạn phát triển kỹ năng của mình với tư cách là người thực hành hoặc người ủng hộ DevOps.

Yêu cầu về phần cứng và phần mềm

Có năm chương đi sâu hơn vào công nghệ. Chương 6, Thực hành kỹ thuật mở - Sự khởi đầu, Quyền bắt đầu và Chương 7, Thực hành kỹ thuật mở - Điểm giữa tập trung vào việc khởi động môi trường kỹ thuật. Chương 14, Build It, Chương 15, Run It và Chương 16, Own It đề cập đến việc phát triển và vận hành các tính năng trong ứng dụng chạy trên nền tảng OpenShift của chúng tôi.

Chúng tôi khuyên tất cả độc giả, bất kể kỹ năng kỹ thuật của họ, hãy khám phá các khái niệm được giải thích trong các chương này. Tùy chọn, bạn có thể muốn tự mình thử một số phương pháp kỹ thuật. Các chương này cung cấp hướng dẫn về cách thực hiện điều đó.

Các yêu cầu về Kích thước OpenShift để chạy các bài tập này được nêu trong Phụ lục A.

Công ước

Các từ mã trong văn bản, tên cơ sở dữ liệu, tên thư mục, tên tệp và phân mở rộng tệp được hiển thị như sau:

Chúng tôi sẽ đề cập đến những điều cơ bản về thử nghiệm thành phần giao diện người dùng PetBattle bằng Jest. Giao diện người dùng được tạo thành từ một số thành phần. Trang đầu tiên bạn nhìn thấy khi truy cập ứng dụng là trang chủ. Đối với thành phần trang chủ, lớp kiểm tra test class is called `home.component.spec.ts`:

```
describe('HomeComponent', () => {
  let component: HomeComponent;
  let fixture: ComponentFixture<HomeComponent>;

  beforeEach(async () => {...
});

  beforeEach(() => {...
});

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

Đang tải xuống tài nguyên

Tất cả các tạo phẩm công nghệ đều có sẵn trong kho GitHub của cuốn sách này tại <https://github.com/PacktPublishing/DevOps-Culture-and-Practice-with-OpenShift/>

Phiên bản có độ phân giải cao của tất cả hình ảnh bao gồm ảnh, sơ đồ và mẫu tạo tác kỹ thuật số được sử dụng đều có sẵn tại <https://github.com/PacktPublishing/DevOps-Culture-and-Practice-with-OpenShift/tree/master/figures>

Chúng tôi cũng có các gói mã khác từ danh mục sách và video phong phú có sẵn tại <https://github.com/PacktPublishing/>. Kiểm tra chúng ra!

Chúng tôi biết rằng công nghệ sẽ thay đổi theo thời gian và API sẽ phát triển. Để biết những thay đổi mới nhất về nội dung kỹ thuật, hãy xem kho lưu trữ GitHub của cuốn sách ở trên. Nếu bạn muốn liên hệ trực tiếp với chúng tôi về bất kỳ vấn đề nào bạn gặp phải, vui lòng nêu vấn đề trong kho lưu trữ này.

Phần 1: Thực hành để hoàn thiện

Trong phần này, chúng tôi sẽ giới thiệu cuốn sách, nó đến từ đâu và nó được tổ chức như thế nào.

Chương 1, Giới thiệu – Bắt đầu với Tại sao tập trung vào mục đích và mục tiêu của cuốn sách khán giả. Chương 2, Giới thiệu DevOps và Một số Công cụ, theo cách nói của chúng tôi, sẽ giải thích DevOps là gì và nó giúp tăng tốc chuỗi giá trị phát triển sản phẩm như thế nào. Chúng ta sẽ khám phá chuỗi này là gì và những điểm nghẽn mà văn hóa và thực tiễn DevOps giải quyết. Chúng tôi sẽ giới thiệu một số công cụ quan trọng mà chúng tôi sẽ sử dụng xuyên suốt cuốn sách để định hướng cách sử dụng nhiều loại thực hành khác nhau mà chúng tôi sắp áp dụng. Trong Chương 3, Hành trình phía trước, chúng tôi sẽ giới thiệu cách chúng tôi sử dụng những câu chuyện trong thế giới thực và nghiên cứu điển hình mà chúng tôi sẽ sử dụng xuyên suốt cuốn sách nhằm phác thảo cách tổ chức sáu phần còn lại của cuốn sách.

Điều này sẽ giúp chúng tôi xây dựng nền tảng và bắt đầu hành trình khám phá, lựa chọn và phân phối liên tục.

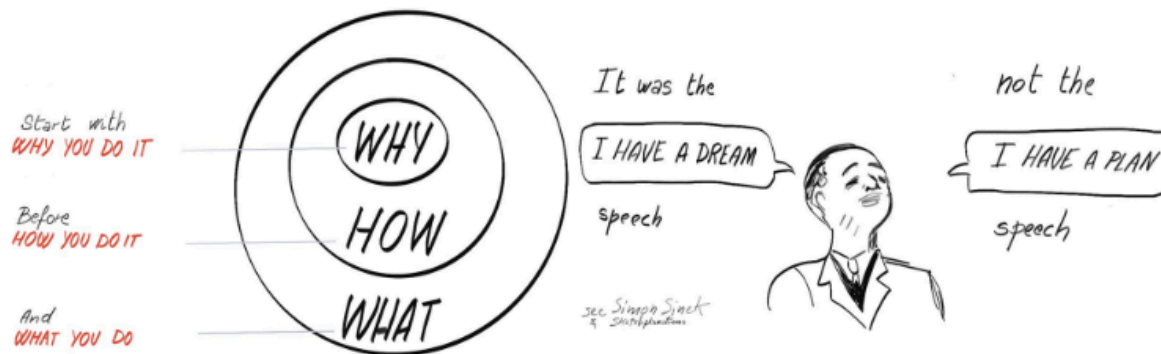
1

Giới thiệu - Bắt đầu với Tại sao

Bạn đã chọn cuốn sách này và bắt đầu đọc nó - cảm ơn bạn rất nhiều!

Có lẽ bạn đã đọc bìa sau và nó cung cấp vừa đủ thông tin để bạn đủ tò mò để mở cuốn sách ra và đọc thêm. Có thể một người bạn hoặc đồng nghiệp đã nói với bạn về điều đó và giới thiệu nó cho bạn. Có thể bạn đã vấp phải nó vì một lý do khác. Dù lý do là gì đi nữa, chúng tôi rất vui vì bạn đã dành chút thời gian trong ngày để bắt đầu đọc cuốn sách này và chúng tôi hy vọng bạn nhận được một số giá trị từ nó và muốn tiếp tục đọc nó.

Trước khi đi vào chi tiết về nội dung cuốn sách này và nội dung của nó, chúng tôi muốn bắt đầu với lý do tại sao. Đây là phương pháp chúng tôi sử dụng để tạo ra tầm nhìn chung về mục đích. Tại sao chúng tôi viết cuốn sách này? Nó đang cố gắng giải quyết những vấn đề gì và đối tượng mục tiêu là ai?



Hình 1.1: Tạo ra tầm nhìn chung về mục đích

Tại sao - Vì lý do hoặc mục đích gì?

Mặc dù cuốn sách này có thể được định vị là một cuốn sách về công nghệ, nhưng nhiều nhất nó chỉ nói về công nghệ một phần ba. DevOps thực sự là về sự cộng tác. Chúng tôi viết cuốn sách này vì chúng tôi muốn nâng cao hiểu biết của bạn về DevOps, cộng tác cũng như các hoạt động văn hóa và kỹ thuật trên nền tảng container như OpenShift. Chúng tôi muốn giúp việc chuyển sang DevOps dễ dàng hơn và cung cấp lộ trình rõ ràng hơn để bạn áp dụng DevOps bằng OpenShift. Chúng tôi muốn khiến bạn phấn khích khi đọc nội dung này và truyền cho bạn một số nguồn cảm hứng về cách bạn có thể áp dụng các nguyên tắc và thực tiễn DevOps. Chúng tôi muốn trang bị cho bạn khả năng ra ngoài và thử những kỹ thuật và phương pháp thực hành mới này.

Khi bạn đọc cuốn sách này, chúng tôi muốn bạn liên tục đo lường tính hữu ích (tác động/giá trị) của việc sử dụng những kỹ thuật mới này. Trên thực tế, mỗi khi bạn thử điều gì đó, chúng tôi muốn bạn suy nghĩ và đo lường tác động của nó.

Tác động đó có thể ở cấp độ cá nhân: Việc thử thứ đó có tác động gì đến tôi, khách hàng hoặc người dùng? Ví dụ: nó có làm giảm thời gian hoàn thành một loạt hoạt động giao hàng của tôi không? Ngoài ra, nó có thể tác động đến nhóm hoặc bộ phận nơi bạn làm việc: Sự hài lòng của nhóm có tăng lên không? Chúng tôi, với tư cách là một nhóm người, đã đạt được điều gì từ điều đó? Tác động thậm chí có thể được cảm nhận ở cấp độ tổ chức hoặc xã hội: Nó có làm giảm số lượng sự cố vận hành ảnh hưởng đến khách hàng không? Chúng tôi tin rằng bạn sẽ nhanh chóng nhận thấy những tác động tích cực ở tất cả các khía cạnh này. Do đó, có thể bạn sẽ để lại cho chúng tôi những đánh giá tốt đẹp và kể cho tất cả bạn bè của bạn về cuốn sách này. Nếu không, có lẽ bạn có thể xoay và sử dụng cuốn sách này như một tấm chặn cửa hoặc một giá đỡ màn hình, tất nhiên, điều này sẽ mang lại cho bạn một loại giá trị khác!

Nếu bạn không biết bắt đầu từ đâu về cách đo lường giá trị, hãy đọc tiếp — chúng tôi hứa sẽ đề cập đến vấn đề đó.

Những gì chúng tôi vừa làm là bắt đầu sử dụng một trong những phương pháp và kỹ thuật được sử dụng để viết cuốn sách này. Chúng tôi đã sử dụng bài thực hành Bắt đầu với lý do tại sao, đây là điều chúng tôi luôn cố gắng thực hiện với mọi nhóm hoặc tổ chức mà chúng tôi làm việc cùng.

Vì vậy, thực hành là gì? Luyện tập là một hoạt động giúp các nhóm đạt được các mục tiêu cụ thể. Đó không chỉ là một ý tưởng; đó là điều bạn làm đi làm lại để trau dồi hoặc trau dồi một kỹ năng. Thực tiễn có các thuộc tính sau:

- Trao quyền: Các phương pháp thực hành trong cuốn sách này sẽ giúp các nhóm khám phá và thực hiện lặp đi lặp lại.
- Súc tích: Chúng có thể được đọc trong vài phút.
- Bất khả tri: Thực tiễn không yêu cầu nhóm phải tuân theo một khuôn khổ cụ thể.
- Đã được chứng minh: Thực tiễn đã được thử nghiệm trong thế giới thực.
- Có thể lặp lại: Các bài tập có thể được sử dụng nhiều lần.

Hy vọng rằng xuyên suốt cuốn sách này, bạn sẽ thấy những ví dụ về việc chúng tôi thực hành những gì chúng tôi giảng thông qua những kinh nghiệm, câu chuyện và lời khuyên mà chúng tôi sẽ chia sẻ từ trải nghiệm truyền đạt trong thế giới thực của mình, bao gồm những câu chuyện như sau:

- Câu chuyện về khi chúng tôi làm việc với một công ty bảo hiểm để xây dựng lại một trong các ứng dụng của họ bằng DevOps và OpenShift nhưng đã có một khoảnh khắc toàn cầu phải dừng lại (một cách thực hành mà chúng tôi sẽ nói đến trong phần tiếp theo) khi chúng tôi nhận ra rằng mình đang phát triển lại một ứng dụng mà người dùng không muốn và không sử dụng!
- Câu chuyện khi chúng tôi làm việc với một công ty ô tô châu Âu và bắt đầu phát triển ứng dụng hiện đại cũng như các phương pháp thực hành linh hoạt với một trong các nhóm của họ, chỉ để chủ sở hữu sản phẩm đặt câu hỏi về cách họ chứng minh với ban quản lý rằng đây là cách làm việc tốt hơn khi việc quản lý chỉ làm việc với bảng tính và con số.
- Câu chuyện về một công ty viễn thông gặp phải tình trạng ngừng hoạt động lớn và gặp sự cố không hoạt động trong thời gian lễ hội và mong muốn tìm hiểu các phương pháp thực hành văn hóa và kỹ thuật mới để thúc đẩy phương pháp tự động mở rộng quy mô và tự phục hồi cho cơ sở hạ tầng và ứng dụng của họ.

Tại sao tôi nên lắng nghe những người này?

Trước khi bạn đọc thêm bất kỳ điều gì từ bốn người viết cuốn sách này, có lẽ bạn nên lùi lại một bước và chia sẻ một chút thông tin cơ bản về nguồn gốc của tất cả các giai thoại, lý thuyết, câu chuyện và lời khuyên của chúng tôi.

Tất cả chúng tôi đều làm việc cho Red Hat. Đặc biệt, tất cả chúng ta đều là thành viên của tổ chức dịch vụ của Red Hat, điều đó có nghĩa là tất cả chúng ta đều thường xuyên tương tác và cung cấp các dịch vụ chuyên nghiệp cho khách hàng của Red Hat. Điều này bao gồm từ việc trợ giúp cài đặt và hỗ trợ áp dụng sớm công nghệ Red Hat cho đến thúc đẩy các chương trình chuyển đổi lớn được củng cố bởi công nghệ Red Hat và văn hóa của Red Hat.

Văn hóa của Red Hat tương đối độc đáo vì nó hoàn toàn dựa trên văn hóa nguồn mở và các tổ chức mở (trong đó Red Hat là một trong những ví dụ lớn nhất). Điều này có nghĩa là tổ chức Red Hat được điều hành theo một tập hợp các đặc điểm phù hợp chặt chẽ với triết lý và văn hóa nguồn mở. Chúng bao gồm sự hợp tác, cộng đồng, tính toàn diện, khả năng thích ứng và tính minh bạch. Chúng tôi thực sự khuyên bạn nên tìm hiểu thêm về triết lý tổ chức mở của Red Hat bằng cách đọc Tổ chức mở¹ của Jim Whitehurst.

Rất nhiều kinh nghiệm đã hình thành nên cuốn sách này cũng như những câu chuyện và lời khuyên mà chúng tôi sẽ chia sẻ bắt nguồn từ những cam kết do Phòng thí nghiệm Đổi mới Mở Red Hat (hay gọi tắt là Phòng thí nghiệm) dẫn đầu. Labs cung cấp một cách tiếp cận phong phú và cởi mở để tạo ra những cách làm việc mới có thể giúp khách hàng và nhóm của họ phát triển các giải pháp kỹ thuật số và tăng tốc giá trị doanh nghiệp bằng cách sử dụng công nghệ mở và văn hóa mở. Dịch vụ chính do Labs cung cấp được gọi là nơi cư trú, là một cam kết có khung thời gian kéo dài từ 4 đến 12 tuần, trong đó các kỹ sư của khách hàng được kết hợp trực tiếp với các chuyên gia văn hóa và công nghệ của Red Hat.

Giữa bốn tác giả, chúng tôi đã tham gia vào hơn 50 phòng thí nghiệm Đổi mới Mở trên khắp thế giới, bên cạnh nhiều cam kết dịch vụ chuyên nghiệp khác. Do tính chất tương đối ngắn của thời gian lưu trú tại Phòng thí nghiệm, chúng tôi có thể học rất nhanh các kỹ thuật khác nhau, các cách tiếp cận khác nhau và các phương pháp thực hành khác nhau. Chúng ta có thể thấy điều gì hoạt động tốt và điều gì không hiệu quả. Chúng tôi có thể xây dựng một bộ sưu tập lớn các câu chuyện và lời khuyên. Cuốn sách này tập trung vào việc chia sẻ những câu chuyện và lời khuyên đó.

Cuốn sách này đến từ đâu?

Tiêu đề của cuốn sách này là sự phát triển của một chương trình hỗ trợ đào tạo mà các tác giả đã phát triển có tên là Hỗ trợ thực hành và văn hóa DevOps. Đây là khóa đào tạo chuyên sâu do Red Hat tổ chức, cung cấp hỗ trợ cho khách hàng, đối tác và nhân viên của Red Hat.

Ban đầu, chúng tôi tạo ra khóa học này vì lĩnh vực dịch vụ của Red Hat mà chúng tôi đang làm việc đang phát triển và chúng tôi cần một cách để tăng cường nhất quán sự nhiệt tình và sự hiểu biết được chia sẻ đằng sau các hoạt động và văn hóa mà chúng tôi đang sử dụng trên toàn cầu, với khách hàng và trong chính chúng tôi, tổ chức. Chúng tôi muốn thực hiện điều này bằng cách khám phá tất cả các phương pháp cơ bản mà chúng tôi đã thấy là đã thành công trong việc đưa nhiều sản phẩm ra thị trường với khách hàng. Điều này bao gồm các phương pháp giúp hiểu lý do và thúc đẩy việc khám phá sản phẩm, cũng như các phương pháp giúp chúng tôi phân phối một cách an toàn, bảo mật và tự tin theo cách lặp đi lặp lại và tăng dần. Và sau đó là kết quả thứ ba, đó là sự vui vẻ. Chúng tôi thực sự không thể hiểu được ý nghĩa của tất cả những điều này nếu bạn không thể vui vẻ, nói đùa và thích thú khi tiếp tục - đó là một trong những thành phần chính của văn hóa ngôn từ bí ẩn đó.

1 <https://www.redhat.com/en/explore/the-open-Organisation-book>

Một trong những yếu tố thành công then chốt đằng sau việc này là việc đưa nhiều kinh nghiệm và câu chuyện đời thực vào quá trình giảng dạy của chúng tôi cũng như sử dụng rất nhiều phương pháp thực hành của chúng tôi đối với bản thân để thực hiện khóa học. Mỗi lần chúng tôi tổ chức khóa học, chúng tôi sử dụng định nghĩa về thực hành done² để giải thích cho người tham gia rằng mọi thực hành mà chúng tôi sắp dạy trong khóa học sẽ được trình bày một cách nhất quán, tuân theo quy trình này:

1. Giới thiệu thực hành với lý thuyết và tổng quan về nó là gì, tại sao bạn nên sử dụng nó và cách sử dụng nó
2. Một bài tập thực hành thực hành để tất cả những người tham gia có thể rời khỏi khóa học sau khi đã thực hành và thu được một số kiến thức cũng như kinh nghiệm từ nó
3. Một ví dụ thực tế về phương pháp đang được áp dụng trong một dự án cung cấp sản phẩm hoặc dự án cung cấp sản phẩm thực tế cho khách hàng

Các phương pháp cốt lõi được giảng dạy trong khóa học này khác nhau, từ các phương pháp khám phá, chẳng hạn như lập bản đồ tác động và tấn công sự kiện, đến các phương pháp phân phối, chẳng hạn như lập kế hoạch chạy nước rút và hồi cứu. Chúng bao gồm một tập hợp các phương pháp mà chúng tôi nhận thấy là rất hiệu quả trong việc thiết lập các nhóm sản phẩm có hiệu suất cao và tồn tại lâu dài, chẳng hạn như hợp đồng xã hội, các phương pháp thực hành tình cảm nhóm cũng như lập trình theo cặp và đám đông. Chúng bao gồm các thực hành kỹ thuật mà nhiều người tham gia khóa học có liên quan chặt chẽ nhất với thuật ngữ DevOps, chẳng hạn như tích hợp liên tục, phân phối liên tục, phát triển dựa trên thử nghiệm và cơ sở hạ tầng dưới dạng mã.

Một trong những khía cạnh độc đáo của khóa học này là sự hấp dẫn của nó đối với nhiều đối tượng. Nó không dành riêng cho các nhà công nghệ hoặc nhà thiết kế. Trên thực tế, chúng tôi đã áp dụng ý tưởng thành lập các nhóm người đa chức năng, từ kỹ sư đến quản lý dự án, từ chuyên gia cơ sở hạ tầng đến nhà thiết kế trải nghiệm người dùng. Chúng tôi cảm thấy khóa học này mang đến cơ hội phá vỡ các rào cản. Chúng tôi cố tình không chạy các tuyến đường khác nhau cho những loại người khác nhau. Mục đích là để những người tham gia có được sự hiểu biết chung về tất cả các phương pháp thực hành có thể áp dụng để thực sự đánh giá cao và hỗ trợ văn hóa DevOps.

Sau khi tổ chức khóa học này hơn một trăm lần trên toàn cầu, chúng tôi đã học được rất nhiều điều từ nó và đã liên tục cải thiện nó trong quá trình thực hiện.

Đối mặt với cơ hội viết một cuốn sách mới về DevOps với OpenShift và áp dụng những bài học mới cũng như các công nghệ cập nhật hơn từ Stefano Picozzi, Mike Hepburn và cuốn sách hiện có của Noel O'Connor, DevOps with OpenShift – Cloud Deployments Made Easy, chúng tôi đã xem xét những thành phần quan trọng để biến DevOps với OpenShift thành công cho bất kỳ tổ chức nào chọn áp dụng công nghệ này.

2 <https://openpracticelibrary.com/practice/defining-of-done/>

Các yếu tố thành công đều dựa trên con người, quy trình và công nghệ thông qua việc áp dụng nhiều phương pháp mà chúng tôi đã sử dụng với khách hàng trên toàn cầu và đặc biệt là các loại phương pháp mà chúng tôi đang giới thiệu và cho phép sử dụng văn hóa DevOps và hỗ trợ thực hành.

Mục đích của cuốn sách này là giúp bạn hiểu và sẵn sàng áp dụng nhiều phương pháp thực hành khác nhau - một số liên quan đến con người, một số liên quan đến quy trình, một số liên quan đến công nghệ - sẽ làm cho văn hóa DevOps và thực tiễn với OpenShift thành công trong tổ chức của bạn.

Chính xác thì cuốn sách này dành cho ai?

Cuốn sách này dành cho nhiều đối tượng - bất kỳ ai quan tâm đến thực tiễn DevOps và/hoặc OpenShift hoặc các nền tảng Kubernetes khác. Một trong những hoạt động đầu tiên mà chúng tôi thực hiện là tập hợp lại và liệt kê những tính cách cũng như kiểu độc giả khác nhau mà chúng tôi dự định viết cho. Chúng bao gồm những điều sau đây:



Hình 1.2: Đối tượng mục tiêu

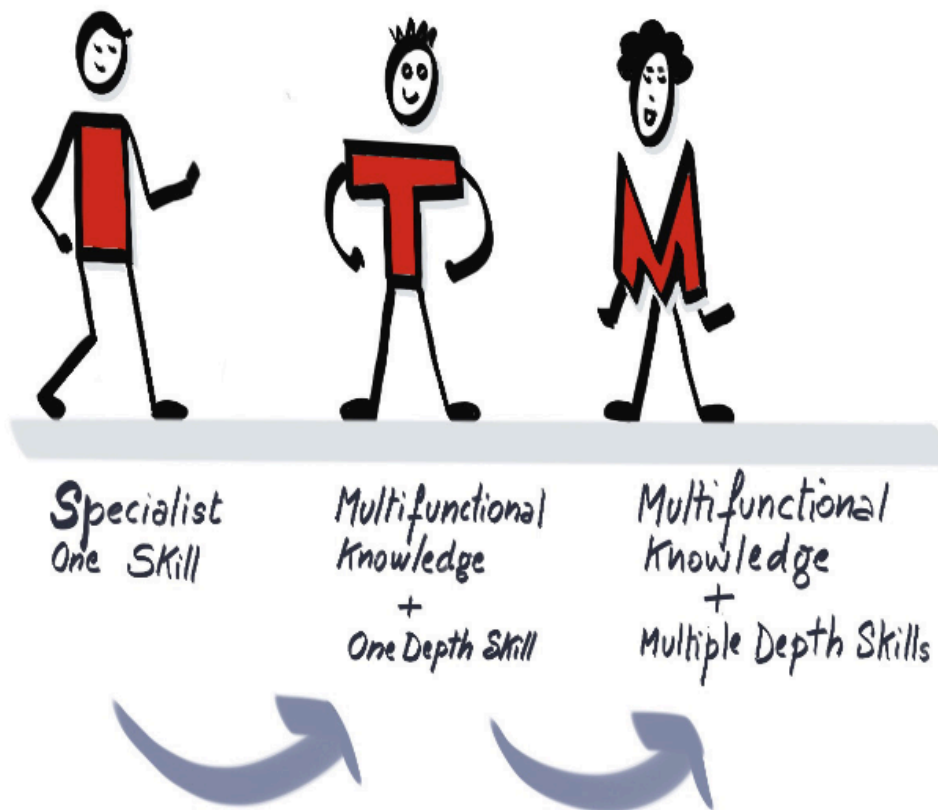
- Caoimhe, trưởng nhóm kỹ thuật, người phụ trách nhóm phát triển phần mềm. Cô ấy muốn tìm hiểu thêm về DevOps để có thể giúp áp dụng các phương pháp thực hành DevOps hiệu quả.
- Fionn, một người quản lý dự án chịu trách nhiệm về một bộ ứng dụng phần mềm cũ và muốn hiện đại hóa cách tiếp cận của nhóm mình để tận dụng điều DevOps mà anh ấy đã nghe nhiều người nói đến.
- Padraig, một huấn luyện viên Agile, người rất có kinh nghiệm trong việc áp dụng các khung phân phối Agile như Scrum và muốn nâng cao kỹ năng cũng như kinh nghiệm của mình với DevOps. Anh ấy cảm thấy rằng điều này sẽ thực sự tăng thêm giá trị cho các đội mà anh ấy đang huấn luyện.
- Tadhg, một nhà thiết kế trải nghiệm người dùng muốn hiểu rõ hơn những gì những người khác trong nhóm phát triển của công ty làm với thiết kế của anh ấy và cách anh ấy có thể cộng tác với họ để phân phối sản phẩm.
- Séamus, một nhà lãnh đạo CNTT đang thực hiện chiến lược công nghệ của công ty mình nhằm áp dụng các bộ chứa và công nghệ dựa trên nền tảng đám mây trên toàn bộ lĩnh vực CNTT của công ty. Anh đã chọn OpenShift Container Platform (OCP) làm sản phẩm chiến lược để hỗ trợ việc này. Anh ấy muốn đảm bảo rằng OCP tạo ra lợi tức đầu tư nhanh chóng và có sự tiếp thu lớn ở tất cả các nhóm CNTT trong tổ chức của anh ấy.
- Aroha, CIO của tổ chức. Cô muốn đảm bảo rằng mọi người trong công ty đều phù hợp với chiến lược của công ty và tận dụng tối đa công nghệ cũng như các quyết định tổ chức được đưa ra để thúc đẩy chiến lược. Cô ấy là động lực để doanh nghiệp trở nên linh hoạt hơn và thích ứng nhanh chóng nếu và khi điều kiện thị trường thay đổi. Cô muốn đọc về những gì các tổ chức có quy mô tương tự trong các ngành khác nhau (kể cả trong ngành của cô) đã thực hiện thành công và những gì họ coi là yếu tố thành công quan trọng.
- Siobhán, một kỹ sư cơ sở hạ tầng đã sử dụng Kubernetes trong nhiều năm và hiện là thành viên nhóm giới thiệu OCP cho tổ chức của cô ấy. Cô ấy muốn đảm bảo rằng nền tảng được cấu hình để hỗ trợ các mục tiêu của nhóm cô ấy và muốn biết cách cô ấy có thể làm việc tốt nhất với các nhóm phát triển để họ nhận được giá trị tối đa từ công nghệ.

- Eimar, người quản lý dự án, người đã dành hai thập kỷ thực hiện các dự án CNTT thông qua việc lập kế hoạch từ trước, theo dõi kết quả thực hiện so với kế hoạch cũng như quản lý rủi ro, vấn đề và sự phụ thuộc bằng kỹ năng báo cáo dự án tốt và quản lý các bên liên quan. Cô cảm thấy thất vọng vì mất nhiều thời gian để vận chuyển phần mềm cũng như không thể giải quyết nhu cầu của người dùng và khắc phục nhanh chóng. Cô nhận thấy lợi ích của việc chuyển sang cách tiếp cận lấy sản phẩm làm trung tâm hơn là lấy dự án làm trung tâm. Cô ấy muốn rèn luyện lại kỹ năng của mình để trở thành người quản lý sản phẩm. Khi làm điều này, cô ấy muốn có thể thử nghiệm và thích ứng nhanh chóng, giao hàng nhanh hơn, thích ứng với những điều kiện thị trường đang thay đổi, đồng thời cải thiện hiệu suất, thời gian hoạt động, thời gian phục hồi, v.v.
- Finn, một người kiểm tra hệ thống rất tự hào về phần mềm đảm bảo chất lượng trước khi nó được chuyển đến khách hàng. Nền tảng phân tích kinh doanh của anh ấy giúp anh ấy phát triển các phương pháp và kịch bản thử nghiệm toàn diện, đồng thời trong nhiều năm, anh ấy cũng đã lãnh đạo thử nghiệm hiệu suất, thử nghiệm bảo mật và thử nghiệm khả năng hoạt động. Anh ấy muốn tìm hiểu cách có thể đưa nhiều tự động hóa hơn vào công việc của mình và mở rộng sang các hình thức thử nghiệm khác.

Từ I đến T đến M

Với cuốn sách này, chúng tôi muốn mọi người thoát khỏi khuôn khổ hình chữ I, nơi họ là chuyên gia trong một kỹ năng hoặc một lĩnh vực. Chúng tôi muốn họ trở thành hình chữ T hơn, nơi họ vẫn có kỹ năng và kinh nghiệm chuyên sâu trong một lĩnh vực cụ thể (chẳng hạn như cơ sở hạ tầng hoặc thiết kế UX), nhưng họ cũng có sự đánh giá cao và kiến thức sâu rộng về tất cả các kỹ năng khác mà mọi người mang lại để tạo thành một nhóm đa chức năng. Ví dụ: đây có thể là một kỹ sư giao diện người dùng, người cũng làm việc cùng với kỹ sư API.

Một nhóm đa chức năng tuyệt vời là một nhóm mà toàn bộ nhóm nắm giữ tất cả các kỹ năng và kinh nghiệm họ cần. Họ được trao quyền để tiếp nhận yêu cầu mới từ người dùng hoặc các bên liên quan trong doanh nghiệp cho đến quá trình sản xuất. Một nhóm có thể được tạo thành từ rất nhiều người hình chữ I, nhưng loại nhóm này nhanh chóng trở nên phụ thuộc vào những cá nhân cụ thể, những người có thể trở thành người cản trở khi họ không có mặt. Ví dụ: nếu cần thay đổi cơ sở dữ liệu để hiển thị API mới nhưng chỉ một thành viên trong nhóm có kiến thức để có thể thực hiện việc này thì nhóm có thể nhanh chóng gặp khó khăn. Nếu nhóm có nhiều thành viên hình chữ T hơn thì sẽ có cơ hội hợp tác, chia sẻ và hợp tác lớn hơn trong toàn nhóm và ít phụ thuộc hơn vào các cá nhân:



Hình 1.3: Chuyển đổi kỹ năng

Chúng tôi muốn cuốn sách này giúp những người có hình dáng chữ I trở nên có hình chữ T hơn và thậm chí có thể trở thành hình chữ M. Những người hình chữ M được truyền cảm hứng để đào sâu việc học, áp dụng nó vào các lĩnh vực khác và nắm giữ nhiều kỹ năng, từ đó xây dựng các nhóm đa chức năng mạnh mẽ hơn.

Phần kết luận

Chương này trình bày tổng quan ngắn gọn về lý do tại sao chúng tôi viết cuốn sách này và nó hướng tới ai.

Chúng tôi đã giới thiệu bản thân và cách chúng tôi sẽ sử dụng kiến thức, kinh nghiệm và kiến thức đã học được để viết cuốn sách đầy những câu chuyện và ví dụ này.

Chúng tôi đã xem xét các tính cách khác nhau mà chúng tôi đang hướng tới trong cuốn sách này và cách chúng tôi dự định giúp chuyển những người tập trung hình chữ I này sang hình chữ T hoặc hình chữ M hơn để xây dựng các nhóm đa chức năng mạnh mẽ hơn.

Trong chương tiếp theo, chúng tôi sẽ giới thiệu DevOps và một số công cụ chúng tôi sẽ sử dụng trong cuốn sách để tổ chức và giải thích các hoạt động DevOps.

2

Giới thiệu DevOps và một số công cụ

Trở thành DevOps trong thế giới container có ý nghĩa gì? Mọi người có những nhận thức khác nhau về DevOps, ý nghĩa và cách thức hoạt động của nó.

Trong chương này, chúng tôi sẽ giải thích quan điểm của chúng tôi về DevOps cũng như những điểm nghẽn và thách thức mà DevOps tập trung giải quyết. Chúng tôi sẽ giới thiệu ý tưởng về chuỗi giá trị trong phân phối sản phẩm phần mềm và cách chúng tôi có thể sử dụng các kỹ thuật khác nhau từ cộng đồng tinh gọn, linh hoạt và DevOps để tối ưu hóa và tăng tốc chuỗi giá trị.

Chúng tôi cũng sẽ giới thiệu một số công cụ, chẳng hạn như Vòng lặp Mobius và Thư viện Thực hành Mở mà chúng tôi sẽ sử dụng để định hướng theo cách của mình thông qua nhiều thực tiễn được sử dụng trong phần còn lại của cuốn sách.

DevOps hiện đang là một từ thông dụng! Có vẻ như cứ mỗi thập kỷ công nghệ lại có một từ thông dụng mới gắn liền với nó.

Trong suốt những năm 2010, Agile là từ thông dụng đó—Đây sẽ là một dự án Agile hoặc Chúng tôi sẽ sử dụng Agile để thực hiện dự án này hoặc Chúng tôi sẽ sử dụng phương pháp Agile là những cụm từ phổ biến mà nhiều người trong chúng tôi đã nghe. Nó đã (và vẫn) thường được sử dụng không chính xác để cung cấp phần mềm nhanh hơn. Trên thực tế, Agile tập trung nhiều hơn vào việc mang lại giá trị kinh doanh sớm hơn và thường xuyên hơn cũng như thúc đẩy văn hóa học hỏi liên tục. Agile hiện đã chính thức trưởng thành — nó đã kỷ niệm sinh nhật lần thứ 18 vào tháng 2 năm 2019. Kể cả sau ngần ấy thời gian, chúng tôi vẫn thích sử dụng các giá trị và nguyên tắc của Tuyên ngôn Agile 1 được tạo ra vào năm 2001.

Container là một từ thông dụng khác ngày nay. Chúng tôi thấy nó đang được các cá nhân sử dụng mà họ không nhất thiết phải hiểu ý nghĩa đầy đủ của vùng chứa là gì và tại sao mọi người, nhóm và tổ chức sẽ được hưởng lợi khi sử dụng chúng.

Vì vậy, với cuốn sách này nói về DevOps và OpenShift (một nền tảng quản lý vùng chứa), chúng ta sẽ làm sáng tỏ các thuật ngữ này và nói về trải nghiệm thực tế, trong thế giới thực cũng như các ví dụ về giá trị thực đằng sau các vùng chứa DevOps và OpenShift.

Chúng ta hãy nhìn lại quá khứ và xem chúng ta tin rằng những hiện tượng này đến từ đâu.

Tất cả chúng ta đều đã làm việc trong lĩnh vực CNTT trong nhiều thập kỷ (nhiều thập kỷ hơn những thập kỷ khác!). Khi trò chuyện bên ly bia và nhìn lại kinh nghiệm thực hiện các dự án CNTT, chúng tôi đã nhận ra một số đặc điểm chung không đổi trong tất cả các dự án CNTT của mình. Chúng tôi cũng xác định được một loạt lỗ hổng trong chuỗi giá trị cung cấp các dự án CNTT mà đối với chúng tôi, dường như chúng đang làm mọi thứ chậm lại.

Chuỗi giá trị

Mọi dự án chúng tôi từng thực hiện đều có một số loại khách hàng hoặc người dùng cuối cùng. Đôi khi họ là người dùng bên ngoài, chẳng hạn như người mua hàng trực tuyến muốn sử dụng ứng dụng di động của họ để mua bộ ga trải giường mới nhất của Justin Bieber! Những lúc khác, họ là các nhóm nội bộ của một tổ chức, chẳng hạn như nhóm vận hành hoặc một bộ phận cụ thể trong công ty. Một mẫu số chung mà tất cả chúng tôi đều đồng ý là mục tiêu công việc của chúng tôi là luôn có những khách hàng vui vẻ, tươi cười:

1 www.agilemanifesto.org



Hình 2.1: Khách hàng hài lòng - Mục tiêu cuối cùng của tổ chức

Giữa chúng tôi, chúng tôi đã giúp đỡ nhiều tổ chức, từ khu vực công và tài chính đến bán lẻ và tổ chức từ thiện. Chúng tôi đã nhìn thấy tất cả! Khi hồi tưởng lại, chúng tôi đã thảo luận về kết quả cuối cùng của một số dự án của mình; chúng tôi đã nghĩ về lý do tại sao của mình - hầu như luôn có một số khía cạnh giá trị tiền tệ liên quan đến lý do chúng tôi có mặt ở đó. Ngoài ra còn có các động lực khác, chẳng hạn như tăng sự hài lòng của khách hàng, giảm rủi ro, cải thiện tính bảo mật và hiệu suất, nhưng điểm mấu chốt là một phần thiết yếu trong hoạt động kinh doanh của bất kỳ khách hàng thương mại nào của chúng tôi là kiếm tiền và giảm chi phí.

Vì vậy, cuối cùng, giá trị thường gắn liền với tiền dưới một hình thức hoặc hình thức nào đó. Ba tác giả của chúng tôi là người Ireland và người thứ tư đến từ New Zealand, vì vậy chúng tôi cảm thấy việc phản ánh điều này như một hũ vàng là điều thích hợp!



Hình 2.2: Lợi nhuận - Mục tiêu chung của mọi tổ chức thương mại

Cuốn sách *The Machine That Changed the World* năm 1990, do James Womack, Daniel Jones và Daniel Roos viết, lần đầu tiên giới thiệu thuật ngữ dòng giá trị. Ý tưởng này còn được phổ biến rộng rãi hơn nhờ cuốn sách *Tư duy Tinh gọn*, được viết bởi chính các tác giả đó. Theo họ, dòng giá trị là chuỗi các hoạt động mà một tổ chức cam kết cung cấp theo yêu cầu của khách hàng. Nói rộng hơn, dòng giá trị là chuỗi các hoạt động cần thiết để thiết kế, sản xuất và phân phối hàng hóa hoặc dịch vụ cho khách hàng và nó bao gồm các luồng thông tin và vật liệu kép. Hầu hết các dòng giá trị đều có tính liên chức năng cao: việc chuyển đổi yêu cầu của khách hàng thành hàng hóa hoặc dịch vụ diễn ra thông qua nhiều bộ phận chức năng hoặc nhóm làm việc trong tổ chức:



Hình 2.3: Khách mơ thấy hũ vàng

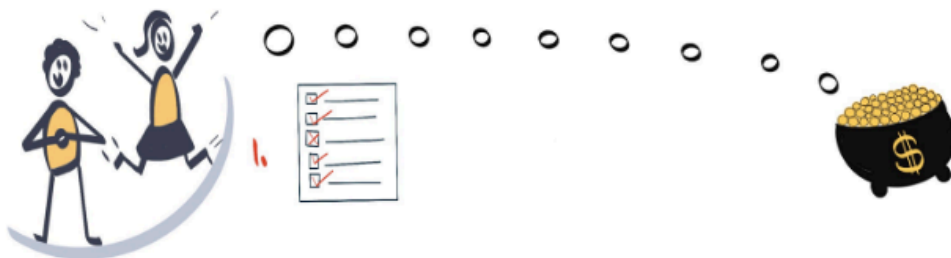
Hãy hình dung điều này giống như khách hàng của chúng ta đang mơ về hũ vàng đó. Họ không ngừng suy nghĩ về cách tận dụng tối đa sản phẩm hoặc ý tưởng của mình để tạo ra nhiều vàng nhất. Vì vậy, làm thế nào để họ thực hiện điều này?

Khoảng trống

Chúng ta sẽ khám phá những lỗ hổng trong chuỗi giá trị giữa khách hàng và nhân viên kinh doanh của tổ chức, giữa nhân viên kinh doanh và nhân viên phát triển cũng như giữa nhân viên phát triển và nhân viên vận hành.

Danh sách lớn những việc cần làm

Lỗ hổng đầu tiên trong quy trình phát triển phần mềm mà chúng tôi liên tục nhận thấy là quy trình thu thập thông tin từ khách hàng cuối và lập danh sách yêu cầu của khách hàng:



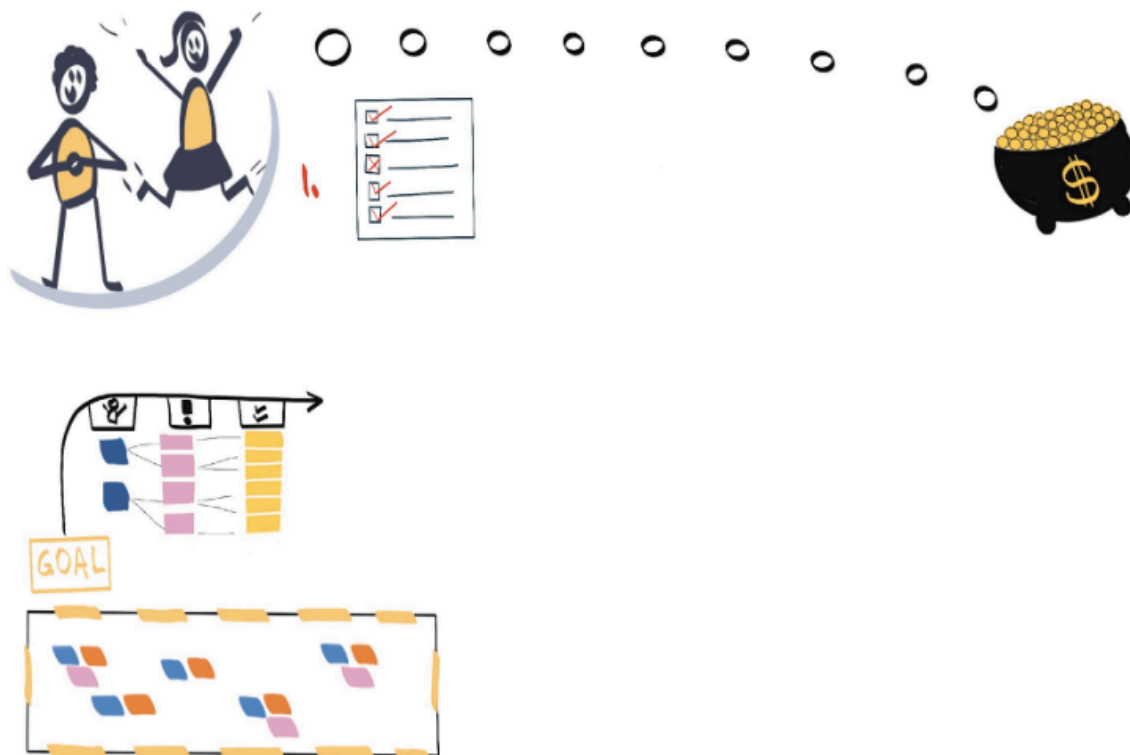
Hình 2.4: Tìm hiểu và thu thập yêu cầu của khách hàng

Các dự án ban đầu của chúng tôi thường bao gồm các giai đoạn dài của các nhà phân tích kinh doanh ghi lại mọi yêu cầu có thể có mà họ có thể hình dung được thành khối lượng lớn các yêu cầu kinh doanh - tài liệu. Mục tiêu là chuẩn bị trước mọi hành trình hoặc tình huống có thể xảy ra của khách hàng và bao quát tất cả các cơ sở bằng cách xây dựng các thông số kỹ thuật bao gồm mọi tình huống có thể xảy ra. Nghe có vẻ cứng nhắc phải không? Điều gì sẽ xảy ra nếu chúng ta đưa ra một giả định không chính xác?

Thể hiện giá trị và xây dựng điều đúng đắn

Khoảng cách thứ hai xoay quanh việc chứng minh giá trị cho khách hàng. Thông thường, dự án đang được thực hiện đã được thiết lập để bao gồm tất cả các tính năng và ý tưởng cần thiết để chúng có thể được phát hành cùng nhau. Khi dự án được đưa vào sản xuất, nó sẽ chỉ có một khoản ngân sách hoạt động nhỏ để hỗ trợ những cải tiến nhỏ và giải quyết vấn đề. Có vẻ như sẽ mất nhiều thời gian để ứng dụng đến tay người dùng cuối phải không?

Có hai lý do chúng tôi gọi những khoảng trống này. Đầu tiên, quá trình này kéo dài - nhiều tháng, đôi khi nhiều năm, từ khi bắt đầu một dự án đến khi ký kết các yêu cầu. Thứ hai, cố gắng thu thập mọi yêu cầu có thể có trước khi cung cấp bất cứ thứ gì có nghĩa là không mang lại lợi ích thực sự cho khách hàng cuối trong nhiều năm và thường, chức năng sai đã được xây dựng và giao cho khách hàng không hài lòng:



Hình 2.5: Sử dụng các phương pháp lấy con người làm trung tâm để tìm hiểu nhu cầu khách hàng

Khoảng cách về việc không xây dựng đúng thứ đã được lấp đầy trong những năm gần đây bởi sự xuất hiện của tư duy thiết kế và thiết kế lấy con người làm trung tâm. Đây là một tập hợp các phương pháp đặt người dùng cuối làm trung tâm trong việc nắm bắt nhu cầu và yêu cầu của sản phẩm.

Chúng tôi thu thập thông tin bằng cách nói chuyện trực tiếp với người dùng và hình thành sự đồng cảm² hơn với họ:

empathy noun



em·pa·thy | \ 'em-pə-thē \

Definition of *empathy*

1 : the action of understanding, being aware of, being sensitive to, and vicariously experiencing the feelings, thoughts, and experience of another of either the past or present without having the feelings, thoughts, and experience fully communicated in an objectively [explicit](#) manner

also : the capacity for this

Hình 2.6: Định nghĩa của Merriam-Webster về “đồng cảm”

Trong cuốn sách này, chúng ta sẽ khám phá cách các kỹ thuật như lập bản đồ tác động, tạo bảo sự kiện và thiết kế lấy con người làm trung tâm có thể hỗ trợ quá trình phát triển phần mềm như thế nào. Chúng tôi cũng sẽ khám phá các phương pháp thực hành khác để giúp chúng tôi xác định các giải pháp và tính năng, đồng thời đảm bảo quan trọng rằng giải pháp đó được kết nối với giá trị doanh nghiệp. Chúng tôi sẽ chỉ ra cách kết hợp các hoạt động nghiên cứu như nguyên mẫu giao diện người dùng và các cải tiến kỹ thuật với thử nghiệm để thông báo các tồn đọng sản phẩm được ưu tiên tốt theo giá trị kinh doanh được phân phối. Chúng tôi sẽ chỉ cho bạn cách sử dụng vừa đủ thông tin có thể giúp bạn hiểu rõ hơn về sản phẩm.

Chúng ta thực hiện những việc trong danh sách của mình như thế nào?

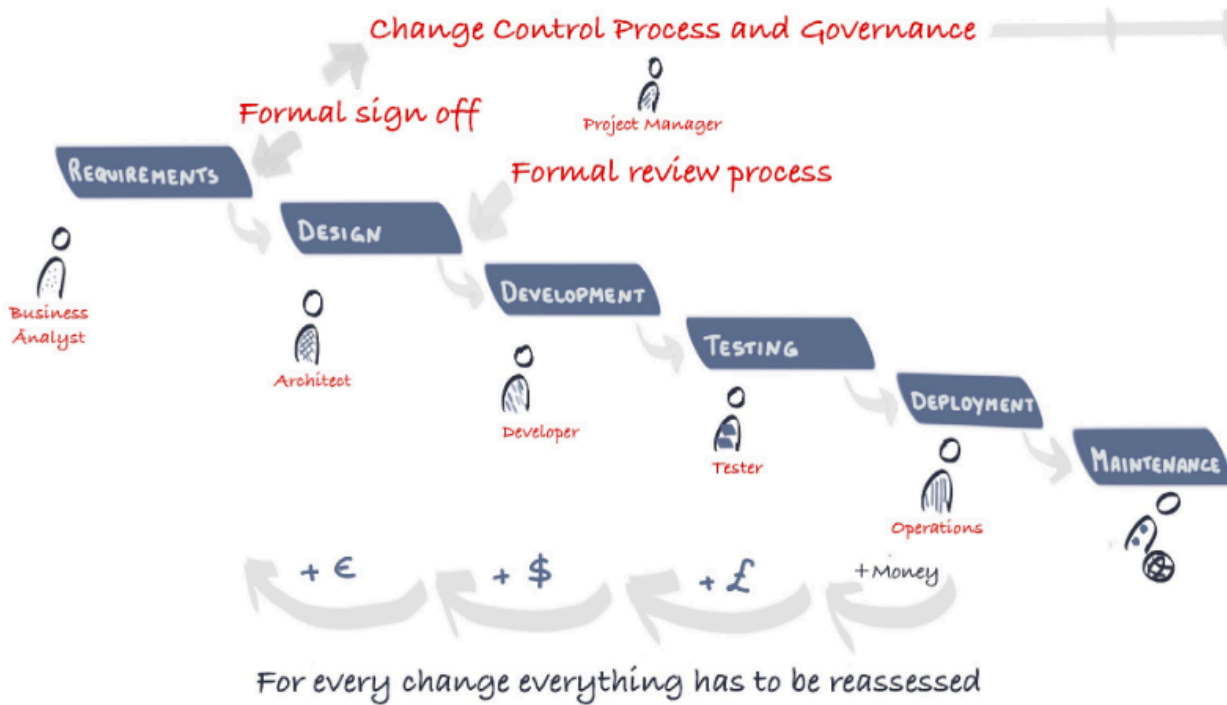
Hãy xem xét lỗ hổng thứ hai trong việc mang lại giá trị cho người dùng. Khoảng cách này tập trung vào việc chuyển từ danh sách mua sắm các hạng mục TODO sang phần mềm hoạt động.

Cách tiếp cận truyền thống là phê duyệt và xác nhận một bộ yêu cầu hữu hạn đã trải qua quá trình phân tích và nắm bắt kinh doanh kéo dài. Phạm vi của dự án được giới hạn và quy trình cũng như quản trị kiểm soát thay đổi nghiêm ngặt được áp dụng để xử lý mọi sai lệch so với các yêu cầu đã được ghi lại.

² <https://www.merriam-webster.com/dictionary/empath>

Sau đó, một nhóm gồm các nhà thiết kế và kiến trúc sư phần mềm sẽ bắt tay vào làm việc, tạo ra một thiết kế cấp cao (HLD) sẽ đưa ra một giải pháp hoặc một bộ giải pháp theo các yêu cầu kinh doanh đã chỉ định. Các yêu cầu này cũng phải trải qua quá trình xem xét chính thức của các bên liên quan chính của dự án và sau khi được phê duyệt sẽ trở thành nguồn tham khảo cho phạm vi giải pháp.

Thông thường, các tài liệu thiết kế khác nhau được viết trong giai đoạn tiếp theo - tài liệu thiết kế chi tiết, thông số kỹ thuật chương trình, thiết kế dữ liệu, bản thiết kế kiến trúc logic, giải pháp kiến trúc vật lý, v.v. Mỗi trong số này được viết để hỗ trợ một phiên bản HLD được xác định, ghi ngày và được phê duyệt, bản thân phiên bản này được ký kết dựa trên một bộ thông số kỹ thuật yêu cầu kinh doanh đã xác định:



Hình 2.7: Vòng đời phát triển ứng dụng truyền thống

Bất kỳ thay đổi nào đối với các tài liệu trước đó đều có tác động trực tiếp đến thời gian và chi phí cho việc đánh giá lại và cập nhật từng tài liệu thiết kế sau đây. Các nhóm phát triển phần mềm có thể đã tham gia vào việc sản xuất hoặc xem xét một số tài liệu này. Tuy nhiên, họ thường được khuyến khích không bắt đầu bất kỳ hoạt động viết mã hoặc phát triển nào cho đến khi những thiết kế này bị khóa. Một số tổ chức giảm chi phí dự án bằng cách không tuyển dụng các nhóm phát triển cho đến giai đoạn này.

Quá trình phát triển thường bị hạn chế bởi chức năng và không nhận thức được bức tranh toàn cảnh với việc thử nghiệm tự động còn hạn chế.

Tại một thời điểm được xác định trước trong kế hoạch dự án, tất cả các nhà phát triển phải chuyển các thành phần được mã hóa của họ đến môi trường thử nghiệm. Có lẽ mỗi nhà phát triển sẽ xây dựng và triển khai mã riêng của họ vào môi trường thử nghiệm theo cách thủ công. Một số chương trình lớn hơn tìm kiếm tính kinh tế nhờ quy mô bằng cách thành lập các nhóm xây dựng cơ sở hạ tầng, những người thực hiện việc này thay mặt cho tất cả các nhà phát triển. Sau khi tất cả các thành phần đã được chuyển giao, một nhóm người thử nghiệm riêng biệt sẽ bắt đầu thực thi hàng trăm tập lệnh thử nghiệm mà họ đã viết trong những tuần và tháng trước đó để thử nghiệm giải pháp theo yêu cầu kinh doanh và tài liệu HLD. Đây là lần đầu tiên một số thành phần được tích hợp và thử nghiệm cùng nhau. Tất nhiên, các vấn đề và lỗi khiến nhóm phát triển và nhà thiết kế phải làm lại để khắc phục những vấn đề đó.

Cũng giống như có nhiều cấp độ khác nhau của tài liệu thiết kế, việc kiểm thử thường trải qua các cấp độ kiểm thử khác nhau, bắt đầu khi giai đoạn trước hoàn thành. Người quản lý kiểm tra sẽ ký vào một tập hợp các kết quả kiểm tra, báo hiệu rằng cấp độ kiểm tra tiếp theo có thể bắt đầu. Thử nghiệm sẽ bao gồm từ một tập hợp thử nghiệm tích hợp thành phần đến thử nghiệm tích hợp hệ thống rộng hơn, thử nghiệm bảo mật và thâm nhập, thử nghiệm hiệu suất, thử nghiệm chuyên đổi dự phòng và khả năng hoạt động, và cuối cùng là thử nghiệm chấp nhận của người dùng!

Giai đoạn cuối cùng trước khi triển khai một giải pháp lớn thường là thử nghiệm mức độ chấp nhận của người dùng, bao gồm một nhóm người dùng trọng tâm và hệ thống thử nghiệm. Trong nhiều trường hợp, thường có thể mất vài tháng hoặc nhiều năm trước khi người dùng đầu tiên này nhìn thấy hệ thống được triển khai. Sau khi người dùng chấp nhận giải pháp, giải pháp sẽ được bật đèn xanh để triển khai vào môi trường sản xuất. Cuối cùng, với phần mềm đến tay người dùng cuối thực sự, hy vọng doanh thu kinh doanh có thể được tạo ra từ tất cả công việc này.

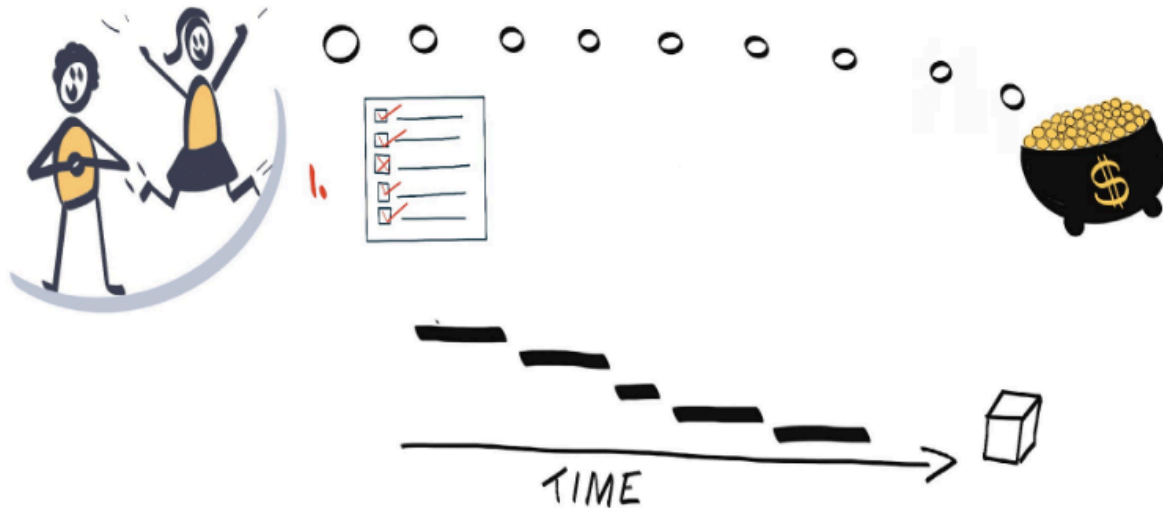
Có thể bạn đang nghĩ rằng quá trình này nghe có vẻ dài dòng và kéo dài - thực ra là như vậy! Nhiều chương trình gặp phải sự chậm trễ ở những thời điểm khác nhau trong quá trình thực hiện và những gì bắt đầu như một kế hoạch dự án kéo dài nhiều tháng cuối cùng lại kéo dài hàng năm. Đối với những người tò mò, thậm chí còn có danh sách một số thất bại hoành tráng trên Wikipedia: https://en.wikipedia.org/wiki/List_of_failed_and_overbudget_custom_software_projects.

Thông thường, điều kiện kinh doanh sẽ thay đổi trong giai đoạn phát triển. Yêu cầu tính năng mới sẽ được tạo ra. Trong quá trình thử nghiệm, những lỗi hổng trong yêu cầu sẽ xuất hiện mà không ai xem xét trong quá trình phân tích và nắm bắt yêu cầu. Thị trường không đứng yên trong quá trình phát triển và các công ty cạnh tranh có thể đã bắt đầu đổi mới nhanh hơn. Sự cạnh tranh thậm chí sẽ cung cấp nhiều yêu cầu tính năng hơn, trong một quá trình giống như một cuộc chiến so sánh tính năng.

Tất nhiên, luôn có một số loại quy trình kiểm soát thay đổi để xử lý phạm vi mới như thế này. Trong một chương trình làm việc phức tạp, thời gian thực hiện để bổ sung các tính năng vào kế hoạch làm việc có thể kéo dài từ vài tháng đến nhiều năm. Để đưa thứ gì đó vào sản xuất, người điều hành chương trình sẽ chỉ cần nói không với bất kỳ thay đổi nào nữa và chỉ tập trung vào việc hoàn thành kế hoạch dự án.

Điều này có nghĩa là các giải pháp cuối cùng được đưa vào sản xuất có phần gây ấn tượng với người dùng vài năm sau khi các yêu cầu đầu tiên được thảo luận. Thời gian và ngành công nghiệp đã tiếp tục phát triển. Sự thất vọng lớn nhất của các chương trình này là chúng thường xuyên được phân phối muộn, vượt quá ngân sách và thường đưa ra giải pháp không làm người dùng hài lòng hoặc thiếu chất lượng.

Lùi lại một chút, chúng tôi gặp phải khoảng cách lớn trong việc chuyển đổi danh sách tính năng thành sản phẩm phần mềm có thể phân phối được. Quá trình được gọi là Thác nước do tính chất của các giai đoạn công việc riêng biệt chuyển sang giai đoạn tiếp theo có liên quan đến thời gian rất dài:



Hình 2.8: Các sản phẩm truyền thống với nhược điểm không đạt được sự hài lòng của khách hàng

Hãy nghĩ về cách chúng ta thu hẹp khoảng cách thứ hai đó bằng các quy trình phát triển phần mềm hiện đại hơn. Làm cách nào để các nhà phát triển hiện đại có thể chuyển nhu cầu của người dùng thành các giải pháp phần mềm hoạt động nhanh hơn nhiều so với các cách làm việc trước đây?

Sự hình thành của phong trào Agile vào năm 2001, do 17 cá nhân CNTT đã viết Tuyên ngôn Agile lãnh đạo, đã khơi dậy những cách tiếp cận và tư duy thay thế đối với việc cung cấp phần mềm. Nhiều cá nhân tham gia viết Tuyên ngôn Agile đã giải quyết nhiều vấn đề được mô tả trong quá trình phát triển Thác nước. Jeff Sutherland và Ken Schwaber đã tạo ra khuôn khổ Scrum để phát triển phần mềm, bao gồm việc cung cấp các bản phát hành nhỏ có giá trị gia tăng thường xuyên hơn - họ sử dụng thuật ngữ chạy nước rút, là một khung thời gian cố định từ một đến bốn tuần (thường là hai tuần), trong đó một tập hợp các sự kiện và vai trò sẽ phối hợp với nhau sao cho các giải pháp lớn có thể được đưa ra lặp đi lặp lại và tăng dần. Kent Beck và Ron Jefferies đã dẫn đầu phần lớn phong trào Lập trình eXtreme (XP), tập trung vào việc cung cấp các bản phát hành giá trị nhanh hơn và nghiên cứu các phương pháp thực hành chính giúp nâng cao hiệu quả trong các quy trình xem xét, thử nghiệm và phát hành, sử dụng khả năng cộng tác tốt hơn và tăng cường tự động hóa:



Hình 2.9: Triển khai thực hành DevOps giúp phân phối nhanh hơn và sản phẩm tốt hơn

Trong cuốn sách này, chúng tôi sẽ giới thiệu cho bạn các phương pháp phân phối phần mềm khác nhau và trải nghiệm của chúng tôi khi sử dụng kết hợp các phương pháp thực hành khác nhau từ Scrum, Kanban, XP, Lean và một số khung mở rộng quy mô giúp mang lại giá trị nhanh hơn như thế nào. Tất cả các thực tiễn cơ bản chỉ đơn giản là công cụ giúp thu hẹp khoảng cách giữa ý tưởng hoặc yêu cầu được nắm bắt và nó được chuyển giao. Đây là lĩnh vực mà chúng tôi đã tìm cách liên tục cải thiện đến mức mà những khoảng cách được giảm thiểu và chúng tôi đang làm việc ở chế độ phân phối liên tục.

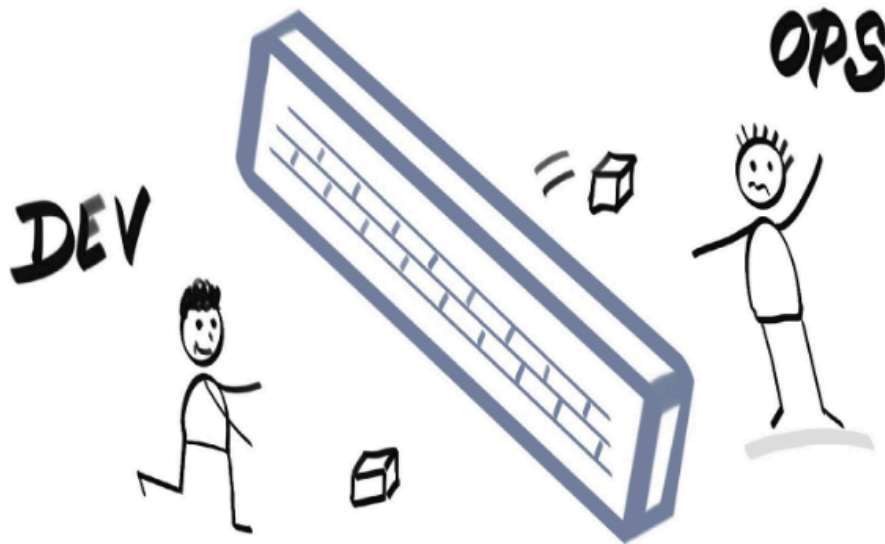
Phát triển đến vận hành

Còn một khoảng trống nữa khiến chúng tôi phải nỗ lực tối ưu hóa quy trình phân phối phần mềm. Khoảng cách thứ ba là khoảng cách giữa nhóm phát triển và nhóm vận hành.

Trong quy trình Thác nước của chúng tôi, chúng tôi đã đạt đến điểm mà giải pháp đã được phê duyệt đã thoát khỏi quá trình thử nghiệm chấp nhận của người dùng và bắt đầu đi vào hoạt động một cách thành công. Vậy điều gì đã xảy ra tiếp theo?

Thông thường, một nhóm hoàn toàn mới chịu trách nhiệm bảo trì và hỗ trợ sẽ đưa ra giải pháp. Những người làm việc trong nhóm mới này không tham gia vào bất kỳ công việc thiết kế, phát triển hoặc thử nghiệm nào, vì vậy thời gian bổ sung sẽ được đưa vào kế hoạch chuyển giao kiến thức của dự án. Nhóm phân phối sẽ viết tài liệu dài với hy vọng rằng đây sẽ là nguồn tài liệu hữu ích cho các nhóm vận hành trong tương lai.

Tại thời điểm này, theo nghĩa ẩn dụ, gói phần mềm sẽ bị ném qua tường từ đội quân các nhà phát triển đến nhóm kỹ sư vận hành. Các nhóm vận hành thường phải tìm hiểu về phần mềm một cách khó khăn bằng cách điều tra các sự cố trong sản xuất, giải quyết các lỗi chưa được tìm thấy trước đó và xử lý các tình huống mới chưa được xem xét trong giai đoạn lập kế hoạch yêu cầu:



Hình 2.10: Mong muốn phá bỏ bức tường giữa nhóm phát triển và vận hành

Để thu hẹp khoảng cách này, chúng ta phải tập hợp các nhóm phát triển và vận hành lại với nhau. Hãy phá bỏ bức tường đó và dỡ bỏ các hầm chứa! Phá bỏ bức tường sẽ hình thành các nhóm mới tập trung vào các hoạt động phát triển và vận hành. Các nhóm này chịu trách nhiệm chung về toàn bộ giải pháp và có thể thiết kế giải pháp theo nhu cầu của nhau.

Thuật ngữ DevOps được đặt ra với ý tưởng rằng chúng tôi không còn có các nhóm vận hành và phát triển riêng lẻ nữa. Trong những năm gần đây, chúng tôi đã thấy nhiều thuật ngữ khác xuất hiện từ ý tưởng này, chẳng hạn như DevSecOps, BizDevOps, DesOps và thậm chí cả BizDesDevSecOps!

Lưu ý BizDesDevSecOps hơi dài dòng, vì vậy chúng tôi sẽ sử dụng thuật ngữ nhóm sản phẩm để mô tả nó trong suốt cuốn sách này. Nó giải quyết mục tiêu cuối cùng là lấp đầy mọi khoảng trống trong quá trình phát triển phần mềm và phá bỏ mọi bức tường.



Hình 2.11: Khắc phục những khoảng trống trong quá trình phân phối phần mềm

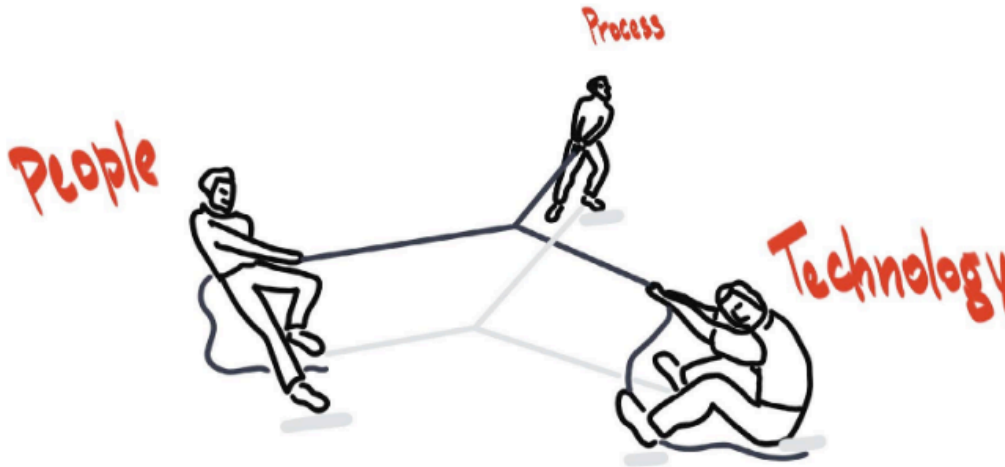
Lưu ý rằng chúng tôi sẽ không sử dụng thuật ngữ nhóm DevOps Nhóm DevOps - ý tưởng về việc có một nhóm hoặc thậm chí một cá nhân hoàn toàn tập trung vào DevOps, đi ngược lại với những gì triết lý DevOps hướng tới - cộng tác, đa chức năng và loại bỏ các silo. Đã bao nhiêu lần bạn nhìn thấy quảng cáo trên LinkedIn hoặc các trang web khác đang tìm kiếm kỹ sư DevOps? Phát minh của kỹ sư DevOps hoặc nhóm DevOps có thể được coi là tạo ra một silo khác.

Con người, quy trình và công nghệ

DevOps thực sự là về sự cộng tác. Đó là sự tự hào và quyền sở hữu giải pháp mà bạn đang xây dựng bằng cách phá bỏ các bức tường và hàm chứa cũng như loại bỏ các nút thắt và trở ngại. Điều này tăng tốc dòng giá trị kết nối nhu cầu nhận thức của khách hàng với việc phân phối sản phẩm.

Chỉ riêng công nghệ sẽ không bao giờ giải quyết được mọi vấn đề trong kinh doanh của bạn. Cho dù nền tảng hoặc sản phẩm phần mềm mà bạn đang đánh giá hoặc bán tốt đến đâu, trừ khi tổ chức của bạn đã học được cách áp dụng sự cân bằng chính xác giữa các khía cạnh con người, thay đổi quy trình và áp dụng công nghệ, nếu không thì các mục tiêu sẽ không đạt được.

Cuốn sách này nói về việc tìm kiếm sự kết hợp phù hợp giữa những thay đổi về con người, quy trình và công nghệ cần thiết để liên tục tối đa hóa kết quả kinh doanh. Điều này đòi hỏi phải thay đổi trong tư duy và thay đổi trong hành vi. Cuốn sách này sẽ xem xét sự thay đổi hành vi mà chúng tôi thấy là hiệu quả nhất với hàng trăm tổ chức mà chúng tôi đã hợp tác chung. Chúng tôi nhận thấy rằng sự thay đổi về tư duy và hành vi như vậy là cần thiết đối với tất cả các vai trò và chúng tôi cần phá bỏ những rào cản mà chúng tôi thấy trong các tổ chức, như chúng tôi đã thấy trước đây, chính là nguyên nhân dẫn đến những khoảng trống và sự kém hiệu quả trong phát triển phần mềm:



Hình 2.12: Sự cân bằng lành mạnh giữa con người, quy trình và công nghệ

Mọi người trong tổ chức nên quan tâm đến con người, quy trình kỹ thuật và công nghệ để đạt được kết quả mong muốn. Chúng tôi muốn phá bỏ rào cản giữa ba trụ cột này và đưa chúng lại gần nhau hơn. Một độc giả có thể quan tâm và tập trung hơn vào một trong ba điều này sẽ nhận được nhiều giá trị (nếu không muốn nói là nhiều hơn) từ việc tìm hiểu về hai điều còn lại.

Điều này có nghĩa là một kỹ sư phần mềm hoặc kiến trúc sư giỏi có thể thu thập được một số hiểu biết sâu sắc và hướng dẫn về lý do tại sao con người, văn hóa và sự cộng tác đều quan trọng như nhau đối với vai trò của họ.

Một người trước đây là chuyên gia về phương pháp quản lý dự án và hiện đang tìm hiểu về các phương pháp phân phối linh hoạt hơn như Scrum cũng có thể sử dụng cuốn sách này để tìm hiểu về các phương pháp công nghệ hiện đại như GitOps, CI/CD và serverless. Họ có thể tìm hiểu lý do tại sao cần hiểu và đánh giá cao những điều này để có thể nói rõ giá trị kinh doanh mà các phương pháp tiếp cận đó mang lại cho tổ chức.

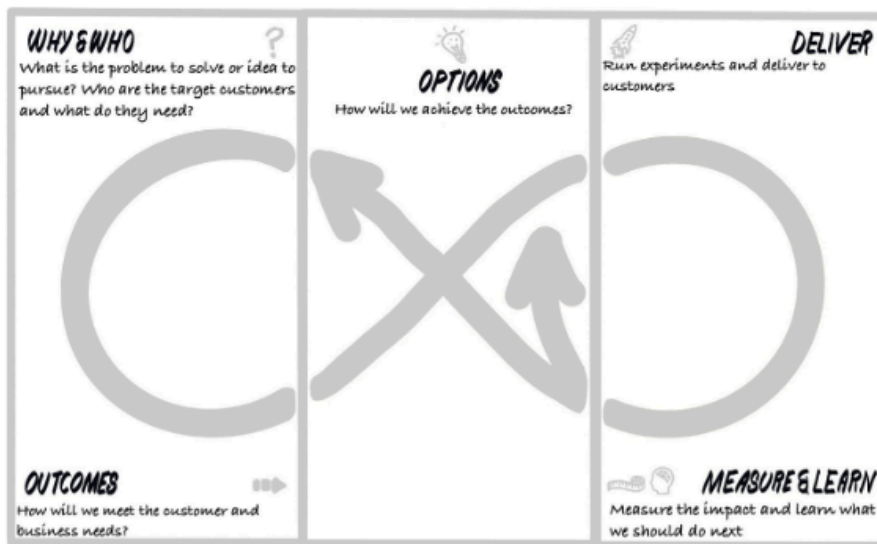
Một nhà lãnh đạo quan tâm đến việc giữ chân nhân viên có thể tìm hiểu làm thế nào việc nắm vững các phương pháp công nghệ hiện đại này của các chiến lược phân phối lặp đi lặp lại và gia tăng có thể tối đa hóa cơ hội thành công của tổ chức thông qua việc cung cấp các sản phẩm có giá trị cao được khách hàng hài lòng sử dụng.

Vòng lặp Mobius và Thư viện Thực hành Mở

Trong cuốn sách này, chúng ta sẽ khám phá rất nhiều cách thực hành khác nhau. Chúng tôi sẽ giải thích chúng là gì và tại sao chúng tôi sử dụng chúng. Chúng tôi sẽ cung cấp cho bạn một số hướng dẫn về cách sử dụng chúng. Chúng tôi sẽ chia sẻ một số ví dụ thực tế về cách chúng tôi đã sử dụng chúng và nếu có thể, chúng tôi thậm chí sẽ hiển thị chúng trong thực tế. Sử dụng nghiên cứu điển hình về Trận chiến thứ cung của chúng tôi (sẽ nói thêm về điều đó sau), chúng tôi sẽ làm cho chúng trở nên sống động theo cách thú vị và chúng tôi sẽ chia sẻ những mẹo hay nhất mà chúng tôi đã thu thập được trong thực tế.

Một vấn đề mà chúng tôi gặp phải cách đây vài năm khi làm việc với nhóm mới của khách hàng là giải thích cách thức và thời điểm bạn có thể muốn sử dụng các phương pháp khác nhau cũng như theo thứ tự nào. Chúng ta nên bắt đầu với sự thực hành nào? Cách thực hành nào liên kết chặt chẽ với kết quả được tạo ra từ cách thực hành trước đó, v.v.?

Để giải quyết vấn đề này, chúng tôi đã sử dụng công cụ điều hướng nguồn mở có tên Mobius. Điều này được tạo ra bởi Gabrielle Benefield và Ryan Shriver. Có một lượng lớn tài liệu tuyệt vời, bao gồm một số bức vẽ và đồ tạo tác có nguồn mở, có sẵn tại www.mobiusloop.com. Phòng thí nghiệm đổi mới mở Red Hat sử dụng tài liệu nguồn mở này trong tất cả các cơ sở lưu trữ cũng như trong các khóa học hỗ trợ thực hành và văn hóa DevOps. 3 Chúng ta sẽ sử dụng nó trong cuốn sách này để cấu trúc nội dung và các phần.



Hình 2.13: Vòng lặp Mobius⁴

3 <https://github.com/rht-labs/enablement-docs>

4 Tài nguyên Mobius Loop (<https://mobiusloop.com/>) của Gabrielle Benefield và Ryan Shriver được sử dụng ở đây và xuyên suốt cuốn sách này được cấp phép theo CC BY 3.0. Ở phần sau của cuốn sách, các hình ảnh bao gồm phần sửa đổi bổ sung của lớp nền. Để biết thêm thông tin, vui lòng xem <https://creativecommons.org/licenses/by/3.0/>

Mobius là một khuôn khổ kết nối việc khám phá và phân phối, đồng thời có thể được sử dụng để kết nối chiến lược, sản phẩm với hoạt động. Mẫu số chung là kết quả có thể đo lường được. Mobius được sử dụng để hiểu, điều chỉnh và chia sẻ các kết quả mục tiêu có thể đo lường được để chúng có thể được kiểm tra và xác nhận.

Có một số nguyên tắc làm nền tảng cho bộ điều hướng Mobius:

- Kết quả hơn kết quả đầu ra: Chúng tôi tập trung vào việc mang lại những tác động hoặc kết quả hữu hình cho mọi người thay vì cung cấp nhiều tính năng có thể không mang lại kết quả.
- Chiến lược nhiều tùy chọn (trực tùy chọn): Chúng tôi xem xét xây dựng danh sách các tùy chọn, danh sách các sáng kiến nghiên cứu, thử nghiệm và tính năng triển khai có thể được sử dụng để kiểm tra các giả thuyết về việc liệu các sáng kiến, thử nghiệm và tính năng triển khai đó có thực sự thúc đẩy hay không những kết quả dự kiến.
- Phân phối nhanh chóng: Chúng tôi mong muốn sử dụng các lần phân phối lặp lại trong thời gian ngắn với phản hồi và đo lường thường xuyên khi chúng tôi nỗ lực hướng tới ý tưởng phân phối liên tục.
- Học hỏi và cải tiến liên tục: diễn ra trong suốt chu trình để các lựa chọn tiếp theo của chúng tôi mang lại tác động thậm chí còn tốt hơn đến kết quả.

Có bảy yếu tố cốt lõi trong cách tiếp cận của Mobius trong một dòng chảy liên tục và không bao giờ kết thúc. Chúng có thể được hiển thị trên một canvas duy nhất là nguồn mở và được cung cấp theo giấy phép Creative Commons tại www.mobiusloop.com:

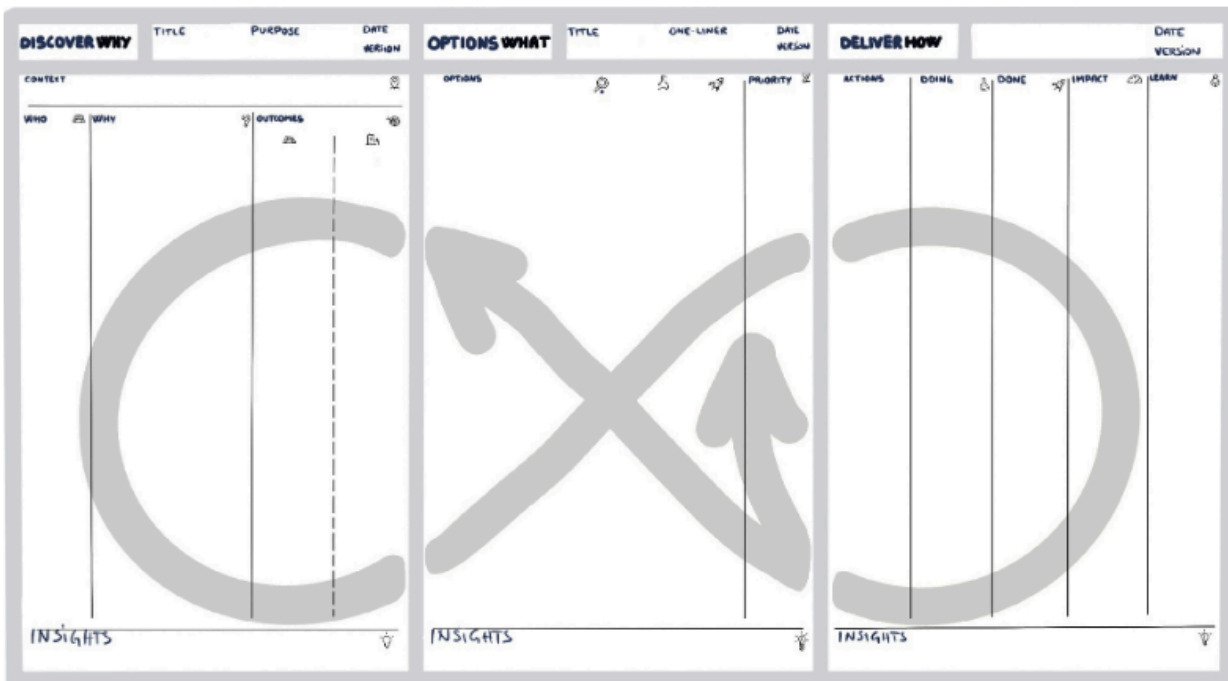
- Tại sao mô tả mục đích. Tại sao chúng ta làm điều này? Vấn đề chúng ta đang cố gắng giải quyết là gì? Ý tưởng mà chúng ta đang cố gắng theo đuổi là gì?
- Ai tập trung vào người dùng cuối. Chúng ta đang cố gắng giải quyết vấn đề cho ai?
- Kết quả là nơi chúng tôi muốn đạt được với những người này, những thay đổi trong hành vi con người của họ có ảnh hưởng đến những kết quả lớn và cách chúng tôi đo lường những tác động mà khách hàng và doanh nghiệp mang lại.
- Các lựa chọn là những giải pháp tiềm năng có thể mang lại những kết quả này. Chúng giúp xác định các giả thuyết mà chúng ta có thể tiếp tục kiểm tra và giúp chúng ta tìm ra cách đơn giản nhất để đạt được kết quả mong muốn với ít nỗ lực hoặc đầu ra nhất.
- Phân phối là chu trình trong đó chúng tôi chạy thử nghiệm để cung cấp giải pháp hoặc bộ giải pháp cho người dùng để chúng tôi có thể đo lường tác động.
- Đo lường là nơi chúng tôi đánh giá những gì đã xảy ra nhờ đưa ra giải pháp hoặc bộ giải pháp. Chúng tôi kiểm tra xem tác động của giải pháp có mang lại kết quả mong muốn hay không và đánh giá mức độ tác động mà chúng tôi đạt được.
- Tìm hiểu là vòng phản hồi đưa chúng ta quay lại trực tùy chọn. Chúng tôi học hỏi từ những gì chúng tôi đã cung cấp và đánh giá những việc cần làm tiếp theo. Chúng ta đã cung cấp đủ để đưa ra đánh giá chưa? Chúng ta có quay lại vòng lặp giao hàng nữa không? Chúng ta đã đạt được kết quả mục tiêu hay những giả định đã học được của chúng ta đã bị vô hiệu hóa chưa? Chúng ta có quay lại vòng khám phá không?

Những nhân vật như Tadhg, nhà thiết kế trải nghiệm người dùng của chúng tôi, thường dành nhiều thời gian cho vòng khám phá. Những nhân vật như Caoimhe, trưởng nhóm kỹ thuật của chúng tôi, theo truyền thống sẽ tập trung vào vòng phân phối. Những nhân vật như Fionn, giám đốc dự án của chúng tôi, thường dành nhiều thời gian ở đây để thiết lập các kết quả và thu thập các phương án. Tuy nhiên, khi chúng tôi tìm cách chuyển sang các nhóm đa chức năng gồm những người hình chữ T hoặc chữ M, chúng tôi thực sự được hưởng lợi từ việc mọi người tham gia vào mọi giai đoạn của vòng lặp Mobius. Và Mobius tạo ra một ngôn ngữ chung dựa trên các kết quả có thể đo lường được mục tiêu.

Bạn có thể áp dụng các nguyên tắc tương tự của tư duy hướng đến kết quả trong việc phân phối chiến lược, sản phẩm và dịch vụ để hỗ trợ các hoạt động kinh doanh và kỹ thuật - chúng ta sẽ quay lại ý tưởng này ở phần sau của cuốn sách.

Mobius mạnh mẽ vì nó có khuôn khổ bất khả tri. Nó tích hợp với nhiều khuôn khổ và phương pháp hiện có mà bạn có thể đã quen thuộc - Scrum, Kanban, tư duy thiết kế, Lean UX, Tạo mô hình kinh doanh, Khởi nghiệp tinh gọn và nhiều khuôn khổ tuyệt vời khác đã xuất hiện trong vài thập kỷ qua. Bạn không cần phải phát minh lại bánh xe hoặc thay thế mọi thứ bạn đã thích và điều đó phù hợp với bạn.

Bạn có thể nắm bắt thông tin chính trên bản đồ khám phá, bản đồ tùy chọn và bản đồ phân phối — tất cả những thứ này đều là các tạo phẩm nguồn mở có sẵn trong Creative Commons tại www.mobiusloop.com:



Hình 2.14: Sử dụng canvas Khám phá, Tùy chọn và Phân phối của vòng lặp Mobius

Khi Phòng thí nghiệm Đôi mới Mở của Red Hat bắt đầu sử dụng Mobius, chúng tôi đã đặt tất cả các biện pháp thực hành của mình xung quanh vòng lặp Mobius. Một số phương pháp thực hành rõ ràng phù hợp với vòng khám phá và đặc biệt là phần cuối Tại sao & Ai của vòng khám phá. Các phương pháp thực hành như lập bản đồ tác động, lập bản đồ từ đầu đến cuối và lập bản đồ đồng cảm rất hữu ích trong việc tìm ra các câu trả lời được đặt ra trong phần này của vòng lặp. Chúng ta sẽ đi vào chi tiết những cách thực hành này trong các chương tiếp theo của cuốn sách này.

Các phương pháp như tạo sự kiện và lập bản đồ câu chuyện của người dùng rất hữu ích trong việc thiết lập và hình dung các kết quả ở phía bên kia của vòng khám phá. Một lần nữa, chúng ta sẽ xem xét các phương pháp này một cách chi tiết và chia sẻ một số ví dụ tuyệt vời về tác dụng của chúng.

Các hoạt động thực tiễn như chạy nước rút thiết kế, khả năng của chúng tôi và cải tiến sản phẩm tồn đọng sẽ giúp xác định và sắp xếp một loạt các tùy chọn có sẵn nhằm cố gắng đạt được kết quả.

Các phương pháp thực hành như lập kế hoạch chạy nước rút sẽ giúp lập kế hoạch và thực hiện việc phân phối sản phẩm dần dần để đạt được kết quả. Chúng ta sẽ khám phá các phương pháp phân phối lặp lại này và cách sử dụng các khung Agile khác nhau với Mobius.

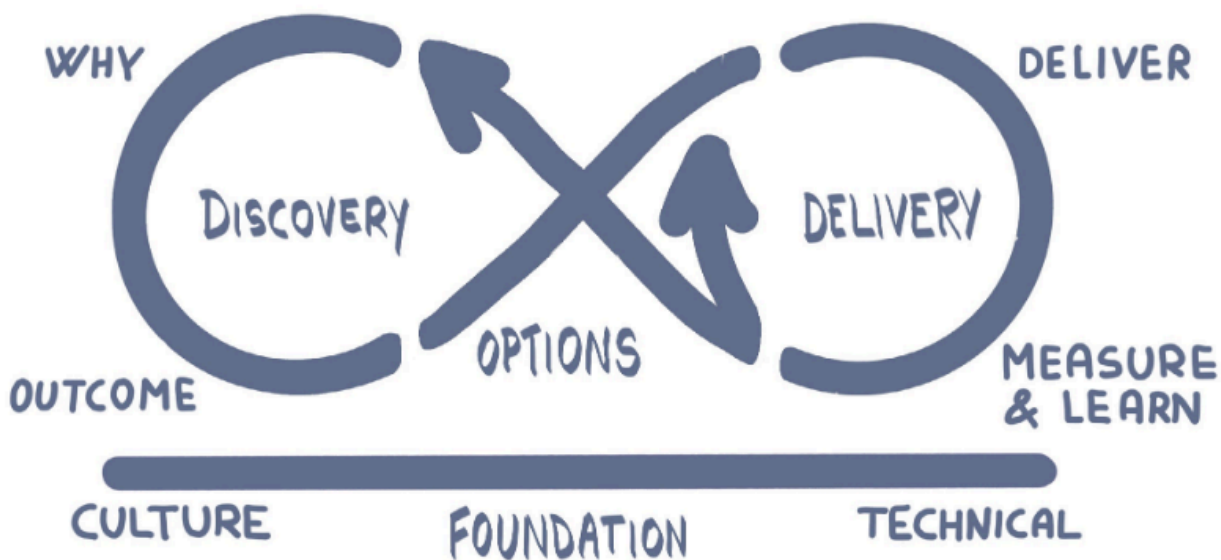
Các phương pháp như trưng bày và hỏi tường sẽ giúp thu thập dữ liệu đo lường và học hỏi từ quá trình phân phối tăng dần.

Chúng tôi vẫn còn một số lượng lớn các phương pháp thực hành mà chúng tôi cảm thấy không phù hợp một cách tự nhiên với một trong các vòng lặp hoặc trực tùy chọn. Khi chúng tôi trình bày một cách hiệu quả tất cả các phương pháp còn lại mà chúng tôi đã áp dụng với nhiều khách hàng, chúng tôi nhận thấy chúng phù hợp với một trong hai lĩnh vực. Tất cả các hoạt động đều tập trung vào việc tạo ra văn hóa và sự hợp tác. Các phương pháp thực hành khác đều là các phương pháp kỹ thuật kỹ thuật hỗ trợ khái niệm phân phối liên tục.

Khi giải thích những phương pháp thực hành này cho người khác, chúng tôi đã nói về việc đây là những phương pháp thực hành rất quan trọng cần thực hiện, nhưng không nhất thiết là những phương pháp thực hành mà bạn sẽ lên lịch. Ví dụ: bạn sẽ biết rằng các phương pháp thực hành như lập bản đồ tác động trên vòng khám phá là những hội thảo quan trọng đã được lên lịch mà bạn thực hiện và thỉnh thoảng xem lại trong tương lai. Các hoạt động như lập kế hoạch chạy nước rút, giới thiệu và cải tiến trong vòng phân phối cũng được lên lịch chặt chẽ khi làm việc trong khuôn khổ phân phối lặp lại. Nhưng những hoạt động liên quan đến văn hóa và cộng tác hoặc những hoạt động liên quan đến kỹ thuật kỹ thuật giống như những hoạt động mà bạn sử dụng mọi lúc, liên tục.

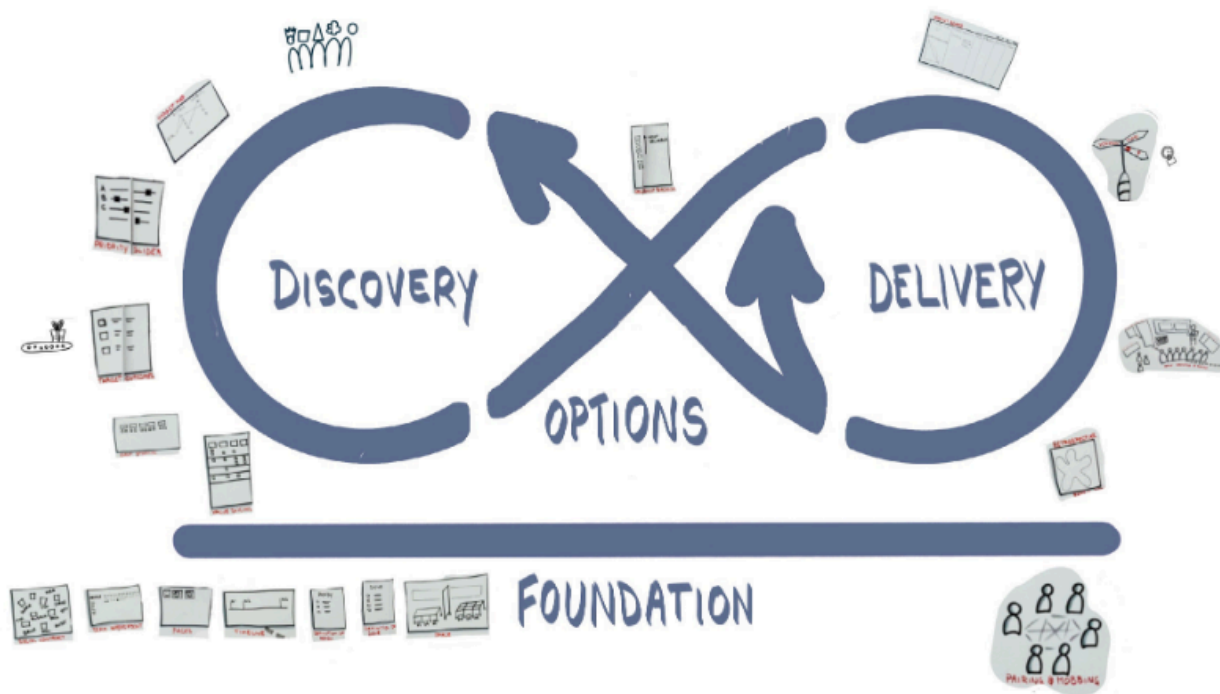
Các hoạt động như hợp đồng xã hội và định nghĩa về việc đã hoàn thành không phải là các hoạt động chỉ thực hiện một lần khi bạn đưa sản phẩm ra ngoài theo lịch trình. Đây là những tạo tác sống và thở mà các nhóm luôn sử dụng trong công việc hàng ngày của họ. Tương tự như vậy, tích hợp liên tục, tự động hóa thử nghiệm và cơ sở hạ tầng dưới dạng mã - đây không phải là những loại thực hành mà bạn lên lịch một hoặc hai lần một tuần. Đây là những thực hành mà bạn làm mọi lúc. Chúng là những thực tiễn làm nền tảng cho địa điểm và cách thức chúng tôi làm việc. Để thực hành hiệu quả việc phân phối liên tục và khám phá liên tục như vòng lặp Mobius trình bày, chúng ta cần có nền tảng vững chắc về văn hóa, cộng tác và thực hành kỹ thuật kỹ thuật.

Để hình dung điều này, chúng tôi đã thêm nền tảng vào vòng lặp Mobius:



Hình 2.15: Thêm nền tảng vào vòng lặp Mobius

Đồ họa này đã trở thành một công cụ trực quan đơn giản giúp chúng tôi điều hướng danh sách các phương pháp và kỹ thuật ngày càng tăng mà chúng tôi sử dụng để đạt được khả năng khám phá và phân phối liên tục các sản phẩm kỹ thuật số:



Hình 2.16: Thực hành khám phá và phân phối liên tục qua vòng lặp Mobius

Open Innovation Labs Residency bao gồm việc di chuyển quanh vòng lặp Mobius một vài lần, thường bắt đầu từ khám phá trước khi tiến hành phân phối và sau đó xoay vòng một vài lần để phân phối nhiều hơn hoặc quay lại khám phá liên tục. Chúng tôi nhận thấy rằng, để điều này được bền vững, bạn phải xây dựng nền tảng văn hóa và cộng tác, đồng thời bạn phải xây dựng nền tảng vững chắc về thực hành kỹ thuật kỹ thuật.

Phòng thí nghiệm Đổi mới Mở đã khởi động một dự án nguồn mở, hướng tới cộng đồng có tên là Thư viện Thực hành Mở. Thư viện Thực hành Mở là kho lưu trữ các thực hành và công cụ hướng tới cộng đồng. Những điều này được chia sẻ bởi những người hiện đang sử dụng chúng hàng ngày dành cho những người muốn tìm cảm hứng từ những ý tưởng và trải nghiệm mới.

Tất cả các thực tiễn bạn đọc trong cuốn sách này đều đã được đóng góp vào Thư viện Thực hành Mở và trong suốt cuốn sách, chúng tôi sẽ sử dụng vòng lặp Mobius và nền tảng của văn hóa, cộng tác và thực hành kỹ thuật làm điểm tham chiếu để xác định vị trí và cách thức tất cả các phương pháp thực hành mở của chúng tôi đều khớp với nhau để mang lại văn hóa DevOps tuyệt vời và phương pháp thực hành với OpenShift.

Một đặc điểm quan trọng của Mobius và Thư viện Thực hành Mở là nó không mang tính quy định. Nó không phải là một phương pháp luận. Nó không cho bạn biết chính xác nên sử dụng phương pháp nào khi nào và ở đâu. Hãy coi Thư viện Thực hành Mở như một hộp công cụ - một hộp công cụ được tổ chức thực sự tốt với nhiều ngăn và kệ. Các hoạt động thực hành đã được tổ chức thành các phần giúp khám phá và đặc biệt là lý do tại sao và ai, sau đó là rút ra kết quả. Có một ngăn kéo chứa tất cả các công cụ giúp hình thành, sắp xếp và ưu tiên các tùy chọn cũng như cách xoay vòng sau này trong chu trình. Có một phần của hộp công cụ chứa tất cả các công cụ hỗ trợ phân phối — cho dù đó là phân phối lặp lại và tăng dần liên quan đến các phương pháp linh hoạt hay phân phối một lần liên quan đến Thác nước. Có các công cụ giúp nắm bắt và hiểu các phép đo cũng như học hỏi từ quá trình giao hàng. Cuối cùng, có rất nhiều công cụ được sử dụng để thiết lập văn hóa, sự cộng tác và kỹ thuật kỹ thuật xuất sắc. Đây thường là những công cụ đầu tiên chúng ta sử dụng khi bắt đầu một công việc.

Phần kết luận

Trong chương này, chúng tôi đã giới thiệu chuỗi giá trị trong phân phối sản phẩm phần mềm và khám phá cách thức làm việc truyền thống mang lại sự thiếu hiệu quả, tắc nghẽn và khoảng cách giữa người dùng, các bên liên quan trong kinh doanh, nhóm phát triển và nhóm vận hành.

Chúng tôi đã khám phá một số kỹ thuật đã được sử dụng để lấp đầy những khoảng trống này và cách tất cả những người liên quan cần tập trung cân bằng vào con người, quy trình và công nghệ.

Cuối cùng, chúng tôi đã giới thiệu công cụ điều hướng nguồn mở có tên Mobius, công cụ này kết nối việc khám phá và phân phối theo một vòng lặp vô hạn, đồng thời có thể kết nối chiến lược với sản phẩm với hoạt động với mẫu số chung là các kết quả có thể đo lường được. Thư viện Thực hành Mở sử dụng mobius trên nền tảng văn hóa và công nghệ để điều hướng giữa số lượng các thực tiễn mở đang ngày càng tăng - nhiều trong số đó sẽ được giải thích trong các chương tiếp theo.

Trong chương tiếp theo, chúng tôi sẽ phác thảo cách tiếp cận phần còn lại của cuốn sách bằng cách giới thiệu nghiên cứu điển hình và cấu trúc của các phần còn lại.

3

Hành trình phía trước

Khi chúng tôi kết thúc phần đầu tiên của cuốn sách này, chương này sẽ giải thích hành trình mà chúng tôi dự định đưa bạn qua các phần còn lại.

Điều này sẽ bao gồm cách chúng tôi dự định không chỉ cho bạn biết về các phương pháp và kỹ thuật mà còn thể hiện chúng trong thực tế và áp dụng chúng. Chúng tôi sẽ giới thiệu một nghiên cứu điển hình thú vị và những câu chuyện thực tế để thực hiện điều này.

Một trong những thách thức khi viết một cuốn sách dành cho nhiều nhóm người có kỹ năng và nền tảng khác nhau là làm thế nào để viết nó theo cách mà tất cả mọi người đều có thể đọc, hiểu và đánh giá cao. Từ các lãnh đạo công nghệ, kỹ sư cơ sở hạ tầng và chuyên gia OpenShift, đến huấn luyện viên Agile, nhà thiết kế trải nghiệm người dùng và quản lý dự án, cho đến lãnh đạo CNTT và CXO, chúng tôi muốn bạn nắm bắt được hiểu biết chung về những gì đang sau tất cả các hoạt động được giảng dạy và các nguyên tắc củng cố sau đó.

Các chủ đề được đề cập sẽ bao gồm từ cách nắm bắt các hành vi trong bản đồ đồng cảm bằng cách sử dụng các phương pháp thiết kế lấy con người làm trung tâm cho đến xem xét khả năng quan sát trong các ứng dụng sử dụng số liệu hiệu suất. Nó sẽ xem xét các cách giúp chủ sở hữu sản phẩm ưu tiên giá trị so với rủi ro, đồng thời giải quyết công cụ đo lường cho ứng dụng, gắn thẻ hình ảnh và siêu dữ liệu!

Tương tự như định nghĩa về thực hành đã thực hiện mà chúng tôi sử dụng trong khóa học hỗ trợ thực hành và văn hóa DevOps, chúng tôi sẽ sử dụng một số phương pháp tiếp cận khác nhau trong cuốn sách này để hỗ trợ bạn trong hành trình của mình:

1. Giải thích văn hóa và tập quán
2. Thể hiện văn hóa và thực hành
3. Áp dụng văn hóa và thực hành

Để giải thích về văn hóa và thông lệ, chúng tôi sẽ giới thiệu thông lệ này là gì, tại sao và ở đâu chúng tôi chọn sử dụng nó, đồng thời đưa ra một số hướng dẫn về cách sử dụng nó. Ở một khía cạnh nào đó, đây là phân dễ dàng.

Giữa chúng tôi có một câu nói mà chúng tôi thích thể hiện hơn là kể ra. Thật dễ dàng để nghiên cứu và viết rất nhiều từ. Sẽ hấp dẫn hơn nhiều khi thể hiện một cách trực quan quá trình thực hành và hiệu quả mà nó mang lại. Để thể hiện văn hóa và thực hành, chúng tôi có một số kỹ thuật:

1. Càng nhiều càng tốt, chúng tôi sẽ hướng tới việc sử dụng các kỹ thuật trực quan hóa như ghi chú phác thảo, sơ đồ và các biểu đồ khác. Bạn sẽ thấy một vài bức trong số này, được vẽ rất đẹp bởi Ilaria Doria, trong phần này, và hy vọng rằng chúng đã giúp ngôn từ trở nên sống động hơn một chút.
2. Khi chúng tôi có thể chỉ cho bạn cách thực hành thông qua các bức ảnh hoặc đồ tạo tác được tái tạo, chúng tôi sẽ làm như vậy. Nếu có thể, chúng tôi đã tạo các sơ đồ và các tạo phẩm trực quan khác thành nguồn mở và chúng có sẵn tại <https://github.com/PacktPublishing/DevOps-Culture-and-Practice-with-OpenShift/>.
3. Chúng tôi nhận thấy những câu chuyện và ví dụ thực tế là cách tốt nhất để giải thích một phương pháp thực hành cũng như giá trị mà nó mang lại. Vì vậy, thỉnh thoảng, chúng tôi sẽ tách ra và kể một câu chuyện mà một hoặc nhiều tác giả đã trải qua liên quan đến những thực hành này. Chúng ta sẽ tự kể những câu chuyện này một cách trực quan bằng cách đặt một khung bao quanh câu chuyện. Hãy bắt đầu với một điều bây giờ:

Câu chuyện kể về một thực hành

Vào tháng 12 năm 2005, tôi đang làm việc trong bộ phận thanh toán của một chương trình thay thế hệ thống thanh toán viễn thông lớn ở Anh. Tính đến thời điểm này, tôi đã làm việc cho chương trình này được 18 tháng. Đó là một chương trình kéo dài 10 năm nhằm thay thế tất cả các hệ thống thanh toán cũ bằng phần mềm COTS hiện đại hơn và giới thiệu một số khả năng kinh doanh mới cho phép quản lý sản phẩm linh hoạt và có thể thay đổi.



Tôi lãnh đạo luồng công việc giao diện thanh toán và chịu trách nhiệm phân phối giao diện giữa hệ thống thanh toán và các bên thứ ba như ngân hàng, BACS và kho bạc.

Quy trình làm việc của chúng tôi là ăn tối Giáng sinh tại một quán rượu gần văn phòng. Chúng tôi chọn quán rượu này vì hầu hết chúng tôi đã từng đến quán rượu này 12 tháng trước vào bữa tối Giáng sinh năm ngoái. Thật buồn cười khi thấy rất nhiều người trong chúng tôi ở cùng một nơi vào cùng một thời điểm trong năm trong 12 tháng qua.

Khi ở đó, tôi đã suy ngẫm về 12 tháng qua và những gì mà đội ngũ tư vấn đắt giá của chúng tôi đã đạt được trong suốt thời gian đó. Tôi chợt nhận ra rằng 12 tháng trước, chúng tôi đang bước vào giai đoạn thiết kế phiên bản chính của chương trình. Chúng tôi đã tổ chức một loạt hội thảo trong vài tuần để vạch ra các sản phẩm công việc khác nhau và các sản phẩm bàn giao cần thiết cho việc phát hành.

Đã 12 tháng trôi qua, chúng tôi vẫn đang trong giai đoạn thiết kế đó. Một nhóm làm việc gồm hơn 60 người đã dành một năm để viết, viết lại, sàng lọc và viết lại các tài liệu thiết kế, nhiều tài liệu thiết kế hơn, các biến thể của tài liệu thiết kế, ghi chú làm rõ kỹ thuật so với tài liệu thiết kế và thậm chí cả các yêu cầu thay đổi đối với tài liệu thiết kế. Tại thời điểm này, chưa có mã nào được

được viết, chưa có bài kiểm tra nào được chạy và không có phần mềm nào được phát hành. Tất cả những gì chúng tôi tạo ra trong 12 tháng là rất nhiều tài liệu thiết kế và rất nhiều cuộc họp.

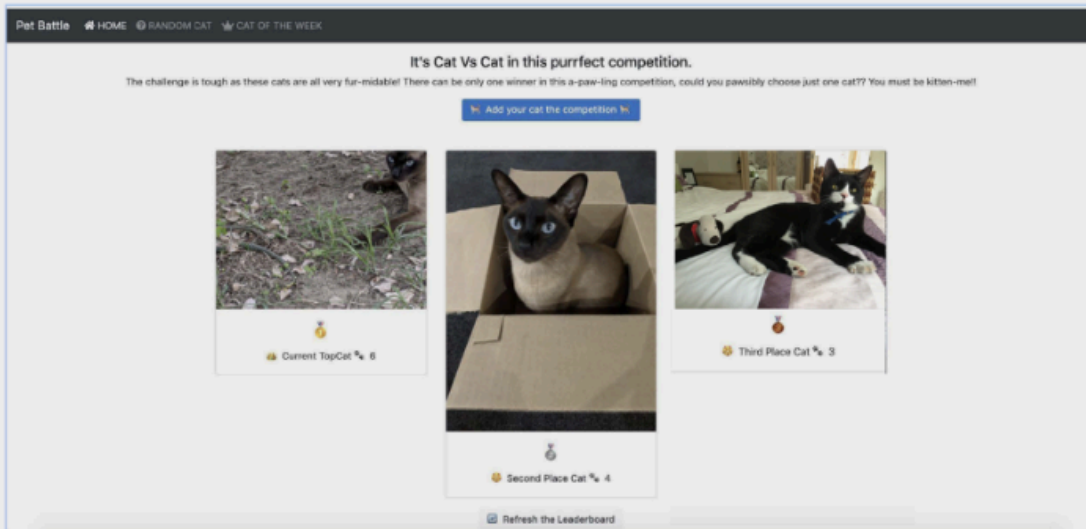
Tôi nhớ cảm giác hơi choáng ngợp trước tác động của những gì chúng tôi đã đạt được trong năm qua. Khi đó tôi tự nhủ: Phải có cách cung cấp phần mềm tốt hơn.

Cuối cùng, chúng tôi muốn áp dụng một số văn hóa và thực tiễn vào thực tế. Để giúp chúng tôi làm điều đó, chúng tôi sẽ sử dụng một nghiên cứu điển hình đơn giản, thú vị về một tổ chức khởi nghiệp nhỏ đang trải qua một số thách thức và trở ngại liên quan đến việc tạo ra văn hóa DevOps và thiết lập các phương pháp thực hành DevOps. Câu chuyện này sẽ trình bày một tài khoản ẩn danh về một số điều chúng tôi đã thấy tại hiện trường với những khách hàng sử dụng các phương pháp này.

Chúng ta sẽ thường xuyên quay lại câu chuyện về việc áp dụng văn hóa và thực tiễn DevOps bằng cách sử dụng OpenShift trong các hộp bóng mờ. Hãy bắt đầu câu chuyện cốt truyện này - chúng tôi hy vọng bạn đã sẵn sàng cho việc này!

PetBattle – Cốt truyện

Hình ảnh của mèo nhà là một trong những nội dung được xem rộng rãi nhất trên internet¹. Điều này có đúng không? Ai biết được! Có lẽ đó là sự thật. Những gì chúng tôi biết là họ tạo ra một cốt truyện tuyệt vời cho ứng dụng mẫu mà chúng tôi sử dụng trong cuốn sách này để giúp giải thích một số phương pháp thực hành DevOps:



Hình 3.1: PetBattle — Cốt truyện

PetBattle là một ứng dụng dành cho người yêu thích, được bắt đầu với mục đích giải trí, được hack xung quanh để các tác giả có thể đấu giữa Mèo và Mèo trong một diễn đàn trực tuyến đơn giản. A Con mèo của tôi tốt hơn loại mèo của bạn. Có rất ít chuông và còi đối với kiến trúc ban đầu - có giao diện người dùng dựa trên web đơn giản và lớp API kết hợp với cơ sở dữ liệu NoSQL.

¹ https://en.wikipedia.org/wiki/Cats_and_the_Internet

PetBattle bắt đầu triển khai trên một máy ảo duy nhất. Nó trực tuyến nhưng không thu hút được nhiều du khách. Nó chủ yếu được bạn bè và gia đình của tác giả thường xuyên lui tới.

Khi đang đi nghỉ ở một thiên đường xa lạ, một trong những tác giả tình cờ gặp một người có ảnh hưởng trực tuyến. Họ hẹn hò, có một kỳ nghỉ lãng mạn và PetBattle đột nhiên trở nên nổi tiếng trên Insta! Gần như chỉ sau một đêm, số lượng người chơi tăng lên đáng kể, máy chủ PetBattle gặp sự cố và những hình ảnh độc hại về những chú mèo không phải mèo bắt đầu xuất hiện trên ứng dụng thân thiện với trẻ em.

Trở về sau kỳ nghỉ, các tác giả đột nhiên thấy mình cần kiếm sống từ PetBattle và quyết định rằng việc phát triển kinh doanh và phiên bản sẵn sàng sản xuất của ứng dụng theo sở thích giờ đây là một việc khả thi.

Bối cảnh được thiết lập để những người sáng lập PetBattle tham gia một cuộc hành trình thú vị theo đuổi văn hóa DevOps và thực hành với OpenShift.

Còn các hệ thống kế thừa thì sao?

Mọi người thường liên tưởng Agile và DevOps với lĩnh vực mới, sự phát triển hoàn toàn mới và hãy xem nó chỉ áp dụng cho những người khởi nghiệp và những người có đủ điều kiện để bắt đầu lại. Còn các hệ thống cũ thì sao? là một câu hỏi phổ biến mà chúng tôi được hỏi.

Trong suốt cuốn sách này, chúng tôi sẽ chỉ ra rằng vòng lặp Mobius và nền tảng có thể áp dụng cho bất kỳ loại dự án và bất kỳ loại công nghệ nào; cánh đồng xanh hoặc cánh đồng nâu, ứng dụng web nhỏ hoặc máy tính lớn, phân phối cơ sở hạ tầng tại chỗ hoặc công nghệ đám mây lai.

Chúng tôi có xu hướng bắt đầu hành trình của mình trên vòng lặp Mobius ở phần khám phá (sau khi xây dựng nền tảng cơ bản về văn hóa, cộng tác và thực hành kỹ thuật). Nhưng bạn không cần phải bắt đầu ở đó. Trên thực tế, bạn có thể bắt đầu ở bất kỳ đâu trên vòng lặp. Mẹo quan trọng nhất là đảm bảo bạn thường xuyên di chuyển quanh tất cả các phần của đường vòng. Đừng mắc kẹt trong các vòng lặp phân phối và đừng bao giờ quay lại khám phá để xem lại các giả thuyết và giả định đã được đưa ra trước đó. Đừng mắc kẹt trong việc khám phá nơi bạn đang di chuyển quá chậm, bạn sẽ bị mắc kẹt trong tình trạng tê liệt phân tích và có nguy cơ bỏ lỡ cơ hội thị trường và không bao giờ mang lại giá trị. Quan trọng nhất, đừng bao giờ quên tiếp tục xây dựng nền tảng văn hóa, hợp tác và thực hành kỹ thuật.

Vay sáng chói

Trước khi bắt đầu đi sâu hơn vào chi tiết, chúng ta nên dành chút thời gian để chỉ ra rằng chúng tôi không viết hay tưởng tượng ra bất kỳ phương pháp thực hành nào trong cuốn sách này. Các thực hành trong cuốn sách này và trong Thư viện Thực hành Mở là danh sách ngày càng tăng những đóng góp của một số bộ óc thông minh. Chúng tôi đã mượn sự xuất sắc đó và sẽ gán nó cho những bộ óc thông minh đã nghĩ ra nó. Chúng tôi hy vọng chúng tôi đã ghi nhận chính xác cho mọi người và mọi thiếu sót đều hoàn toàn là ngẫu nhiên.

Những gì chúng tôi cố gắng làm với cuốn sách này là cho thấy những thực tiễn này, khi được kết nối với nhau, đã mang lại một số kết quả có tác động lớn như thế nào cho các tổ chức và chỉ ra một số nguyên tắc cơ bản cần thiết để tạo ra những kết quả đó.

Những gì mong đợi từ phần còn lại của cuốn sách này?

Vậy là bạn đã gần như hoàn thành xong phần đầu tiên này. Cảm ơn vì đã gắn bó với chúng tôi cho đến nay! Chúng tôi hy vọng bạn cảm thấy hứng thú và có động lực để đọc thêm và có đủ niềm tin vào chúng tôi để muốn đọc những gì chúng tôi đã viết.

Nếu bạn chỉ cần thêm một chút thông tin về những gì sẽ xảy ra thì đây là phần tổng quan ngắn gọn.

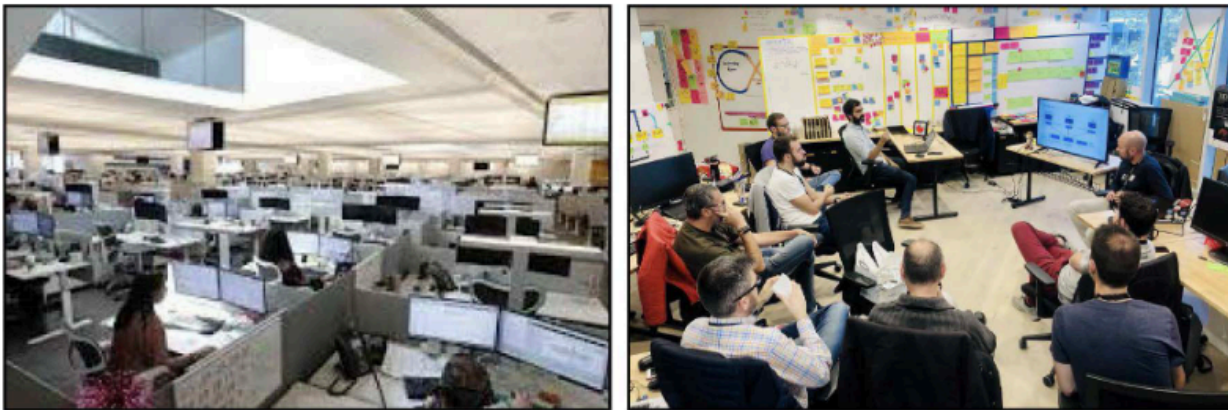
Phần 2 - Thành lập Quỹ

Trong phần này, chúng ta sẽ xem xét sâu hơn tầm quan trọng của việc thiết lập nền tảng cả về mặt văn hóa và kỹ thuật. Chúng ta sẽ xem xét lại động cơ mục đích - sự khởi đầu với lý do tại sao chúng ta bắt đầu cuốn sách này và tại sao đó sẽ là điểm khởi đầu cho bất kỳ sản phẩm hoặc bất kỳ nhóm nào. Chúng ta sẽ xem xét một số phương pháp thực hành được yêu thích và hiệu quả nhất mà chúng ta đã sử dụng để giúp tạo ra nền tảng và văn hóa cộng tác — hợp đồng xã hội, dây andon ngăn chặn thế giới, hồi cứu thời gian thực, tạo bản sắc nhóm và tham gia thói quen hình dung mọi thứ và có một chu kỳ kiểm tra và thích ứng. Sự tập trung không ngừng vào việc tạo ra một môi trường an toàn về tâm lý là yếu tố thành công then chốt khi thiết lập nền tảng. Chúng tôi sẽ giải thích đây là gì và làm thế nào chúng tôi có thể giúp đạt được nó.

Chúng ta sẽ khám phá cách tài trợ điều hành có thể kích hoạt và cản trở một nền tảng thành công và khám phá sâu hơn ý nghĩa của việc mở về mặt công nghệ và văn hóa. Chúng ta sẽ xem xét cách thức hoạt động của việc ra quyết định theo Agile cũng như một số công cụ và phương pháp thực hành hữu ích có thể trợ giúp việc này. Và chúng ta sẽ xem xét cách tiếp cận áp dụng và cách thuyết phục những người nghi ngờ và hoài nghi!

Từ góc độ nền tảng kỹ thuật, chúng tôi sẽ chia sẻ một số phương pháp tiếp cận thành công nhất của chúng tôi, bao gồm trực quan hóa công nghệ thông qua một bức tranh lớn, triết lý xanh từ cờ vây và cách chúng tôi xử lý mọi thứ dưới dạng mã. Chúng tôi sẽ giới thiệu một số số liệu cơ bản mà chúng tôi đã sử dụng để đo lường sự thành công và tác động của văn hóa và thực tiễn DevOps. Chúng tôi thậm chí sẽ tạo bối cảnh cho một số sự đánh đổi và phương pháp thực hành kỹ thuật cần cân nhắc khi tạo nền tảng của bạn — GitFlow — so với phát triển dựa trên Trunk, thiết lập quy trình phát triển, xem xét các loại thử nghiệm khác nhau và thiết lập môi trường để ghép nối và quấy rối.

Để hiển thị chứ không phải để nói, việc thiết lập nền tảng là biến hình bên trái thành hình bên phải:



Hình 3.2: Hợp tác trong tổ chức

Phần 3 - Khám phá nó

Ở đây, chúng ta sẽ đi sâu vào vòng khám phá của Mobius và xem xét một số cách tốt nhất để sử dụng nó. Chúng tôi sẽ chia sẻ một số thực tiễn yêu thích và có tác động mạnh nhất của chúng tôi từ Thư viện Thực hành Mở đã giúp chúng tôi trong vòng khám phá, bao gồm lập bản đồ tác động, thiết kế lấy con người làm trung tâm và tổ chức sự kiện.

Chúng ta sẽ xem xét điều này liên quan như thế nào đến công nghệ và ý tưởng về kiến trúc mới nổi cũng như cho phép phân phối liên tục thực sự.

Từ góc độ kinh doanh, chúng ta sẽ khám phá sự khác biệt giữa kết quả và kết quả đầu ra cũng như cách chúng ta đang cố gắng chuyển từ ý tưởng có nhiều tính năng hơn sẽ tốt hơn sang tạo ra kết quả mạnh mẽ với ít tính năng hơn. Chúng ta sẽ khám phá một số phương pháp thực hành về cách chúng ta có thể liên tục đo lường kết quả và cách chúng ta có thể truyền tải thông tin từ toàn bộ vòng khám phá trên các khung vẽ nguồn mở.

Để hiển thị chứ không phải để nói, chúng ta sẽ xem xét việc chuyển khám phá từ trông giống như những gì bạn thấy ở bên trái sang những gì bạn thấy ở bên phải:

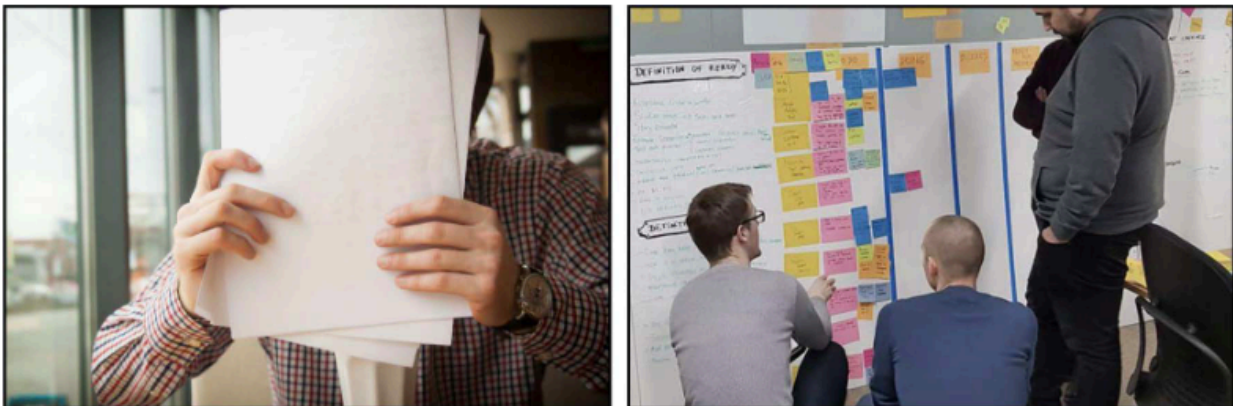


Hình 3.3: Thực hành khám phá thông qua lập bản đồ tác động, thiết kế lấy con người làm trung tâm và tấn công sự kiện

Phần 4 - Ưu tiên nó

Tại đây, chúng ta sẽ đi sâu vào trực tùy chọn của Mobius và xem tại sao các lựa chọn sống, thờ và luôn thay đổi lại quan trọng. Chúng tôi sẽ khám phá các phương pháp thực hành như lập bản đồ câu chuyện của người dùng và phân chia giá trị giúp chúng tôi giải quyết vấn đề này, đồng thời chia sẻ một số câu chuyện đáng tiếc mà chúng tôi có về việc điều này đã bị hiểu sai và sử dụng sai mục đích. Chúng ta sẽ xem xét cách chúng ta xây dựng hồ sơ tồn đọng sản phẩm ban đầu đó bằng cách sử dụng khám phá dẫn đến các phương pháp thực hành xoay vòng tùy chọn. Chúng ta sẽ xem xét các loại hạng mục khác nhau nằm trong sản phẩm tồn đọng, bao gồm từ công việc nghiên cứu đến công việc thử nghiệm và công việc triển khai. Chúng ta sẽ xem xét một số mô hình ưu tiên kinh tế và cách đánh giá sự cân bằng giữa giá trị và rủi ro bằng tư duy thử nghiệm liên tục và học hỏi liên tục. Chúng tôi có rất nhiều câu chuyện để chia sẻ - một số câu chuyện tập trung vào một lĩnh vực cụ thể và một số câu chuyện liên quan đến việc học hỏi trên nhiều lĩnh vực.

Để hiển thị chứ không phải để nói, chúng ta sẽ xem mức độ ưu tiên có thể chuyển từ trông giống như những gì ở bên trái sang những gì ở bên phải:



Hình 3.4: Sử dụng trực Tùy chọn để ưu tiên các mục tồn đọng

Phần 5 - Cung cấp nó

Trong phần này, chúng ta sẽ xem xét phân phối Agile cũng như áp dụng phương pháp này ở đâu và khi nào tùy theo mức độ phức tạp và đơn giản. Chúng ta cũng sẽ xem xét Thác nước và những ưu điểm tương đối cũng như xem nó có thể phù hợp ở đâu. Chúng ta sẽ khám phá các khung linh hoạt khác nhau hiện có và cách tất cả chúng liên quan đến Thư viện thực hành mở và vòng lặp Mobius. Chúng ta sẽ khám phá tầm quan trọng của việc trực quan hóa cũng như việc nắm bắt các phép đo và học tập. Về mặt công nghệ, chúng ta sẽ xem xét cách các kỹ thuật triển khai nâng cao hiện có giúp củng cố một số phương pháp thử nghiệm và học tập đang được thúc đẩy.

Để hiển thị chứ không phải cho biết, chúng ta sẽ xem cách nhận hàng từ trông giống như hình bên trái đến giống như hình bên phải:



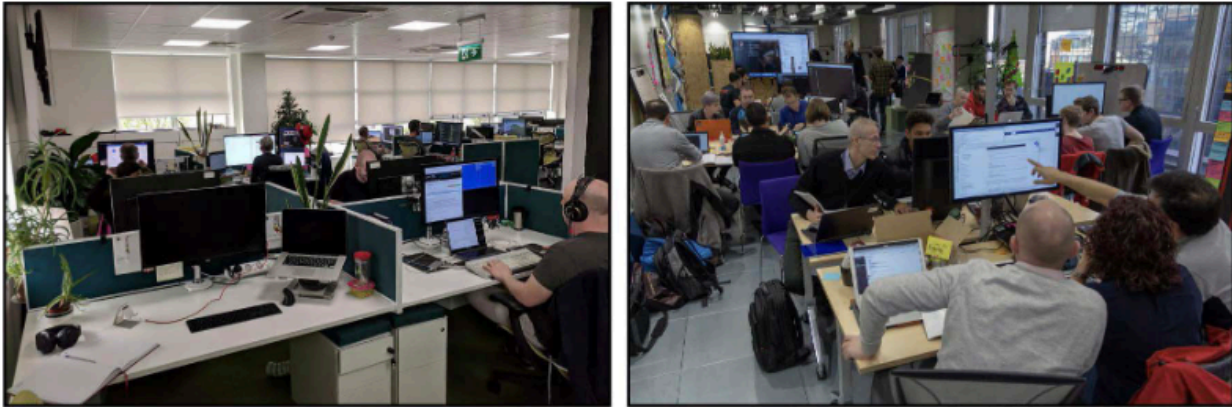
Hình 3.5: Thực hành phân phối thông qua trực quan hóa và đo lường

Phần 6 - Xây dựng nó, vận hành nó, sở hữu nó

Phần này thực sự tập trung vào công nghệ như một yếu tố hỗ trợ và tại sao việc có nền tảng ứng dụng lại quan trọng.

Chúng ta sẽ quay lại triết lý mọi thứ đều dưới dạng mã và xem Git và Helm là những yếu tố hỗ trợ cho việc này. Chúng ta sẽ tìm hiểu sâu hơn về các vùng chứa và hệ sinh thái dựa trên nền tảng đám mây (đám mây, nền tảng và vùng chứa). Chúng ta sẽ khám phá OpenShift và Cloud IDE, cũng như các quy trình cho phép tích hợp liên tục, bao gồm Jenkins và Tekton. Chúng ta sẽ khám phá các phương pháp triển khai và cấu hình mới nổi, chẳng hạn như GitOps thông qua ArgoCD, cùng với hướng dẫn về cách thức và vị trí lưu trữ cấu hình. Chúng ta sẽ khám phá các kỹ thuật triển khai nâng cao, chẳng hạn như thử nghiệm A/B, chuyển đổi tính năng, triển khai canary và triển khai xanh/xanh cũng như cách sử dụng các kỹ thuật này với thử nghiệm kết quả kinh doanh. Chúng ta sẽ xem xét các khía cạnh phi chức năng của thực tiễn DevOps, bao gồm Tác nhân chính sách mở (OPA), quét hình ảnh, DevSecOps, BaseImage và xây dựng chuỗi. Chúng ta sẽ xem xét một số thử nghiệm chức năng và phi chức năng. Chúng ta sẽ khám phá các khía cạnh hoạt động như khung ứng dụng, gắn thẻ hình ảnh, siêu dữ liệu và công cụ gắn nhãn, Knative và serverless cũng như khả năng quan sát về mặt kinh doanh so với các chỉ số hiệu suất ứng dụng. Chúng tôi sẽ tham khảo Lưới dịch vụ và tập trung vào các nhà khai thác để cân nhắc việc quản lý và vận hành ngày thứ 2.

Để hiển thị chứ không phải để nói, chúng ta sẽ khám phá việc xây dựng và vận hành từ những gì bạn thấy ở bên trái sang những gì bạn thấy ở bên phải:



Hình 3.6: Tạo môi trường phù hợp để thực hiện DevOps

Phần 7 - Cải thiện nó, duy trì nó

Khi bước ra khỏi vòng phân phối, chúng tôi sẽ hỏi, Chúng ta đã học đủ chưa? Chúng ta nên xoay vòng hay đi vòng lại vòng phân phối? Chúng ta sẽ thấy chúng ta đang bước vào một chu trình học tập liên tục như thế nào – không phải là việc chỉ diễn ra một lần. Các giả định được chứng minh hoặc bị bác bỏ trong các vòng phân phối. Chúng tôi khám phá thế giới nợ kỹ thuật và cách chúng tôi có thể đưa các số liệu định tính và định lượng từ nền tảng, tính năng và quy trình làm việc của nhà phát triển ứng dụng để giúp giải quyết vấn đề này. Chúng tôi sẽ tìm cách thực hiện các phép đo và học hỏi từ quá trình phân phối trở lại các hiện vật khám phá, chẳng hạn như các cơn bão sự kiện, bản đồ quy trình dựa trên số liệu và nghiên cứu người dùng.

Chúng ta sẽ tìm hiểu cách kết hợp mọi thứ trong vòng khám phá, trực tùy chọn, vòng phân phối và nền tảng để giúp duy trì cách làm việc này. Đây là điều cho phép học theo vòng lặp kép để liên tục khám phá và phân phối liên tục.

Các nhóm sản phẩm đa chức năng, tồn tại lâu dài sẽ học cách xây dựng, vận hành và sở hữu nó. Trong phần này, chúng ta sẽ xem xét một số phương pháp giúp họ duy trì nó.

Vai trò lãnh đạo đóng vai trò gì trong tất cả những điều này? Chúng tôi sẽ trình bày cách trực quan hóa mối liên hệ giữa chiến lược lãnh đạo, phát triển sản phẩm và vận hành nền tảng, tất cả đều được thúc đẩy bởi mục đích và được cung cấp thông tin bằng thông tin và số liệu.

Chúng ta sẽ khám phá những cách tiếp cận để mở rộng quy mô mọi thứ được mô tả trong cuốn sách và tại sao văn hóa tuân theo các nguyên tắc lại quan trọng hơn việc sử dụng các thực hành tôn giáo thuần túy.

Còn các Nhóm phân phối thì sao?

Khi nhìn vào những bức ảnh ở phần trước, bạn có thể nhận thấy rằng thế giới mà chúng ta đang hướng tới bao gồm sự hợp tác của mọi người. Chúng tôi ở trong cùng một không gian, xung quanh những tấm bảng trắng lớn với rất nhiều tờ giấy dán đầy màu sắc.

Một câu hỏi phổ biến mà chúng tôi nhận được từ các nhà lãnh đạo, giám đốc điều hành và khách hàng là việc áp dụng các phương pháp thực hành này như Bảo tố sự kiện hoặc Hợp đồng xã hội khi mọi người không ở cùng một địa điểm có dễ dàng không.

Trong vài thập kỷ qua, các mô hình phát triển ngoài khơi và gần bờ đã gia tăng đều đặn. Một số tổ chức đã được cơ cấu lại dẫn đến các bộ phận khác nhau của tổ chức được đặt tại các khu vực nhất định trên thế giới. Trong một số trường hợp, điều này dẫn đến một tổ chức im lặng hơn với những bức tường lớn hơn giữa các bộ phận khác nhau do địa lý, múi giờ và thiếu sự cộng tác trực tiếp.

Câu trả lời của chúng tôi về việc liệu con người, quy trình và thực tiễn công nghệ mà chúng tôi sắp khám phá trong cuốn sách này có thể được sử dụng với các nhóm phân tán hay không là có, họ có thể.

Tuy nhiên, tốc độ mà một nhóm phân tán sẽ khám phá, thực hiện và học hỏi rất khó có thể nhanh bằng tốc độ làm việc của cùng một nhóm ở cùng địa điểm. Khả năng học hỏi và học hỏi nhanh là nền tảng cho toàn bộ cách làm việc. Vì vậy, chúng tôi luôn tìm kiếm cơ hội để tìm ra cách học hỏi nhanh nhất và loại bỏ những điểm nghẽn có thể làm chậm quá trình học tập của nhóm. Hoạt động phân tán trong hầu hết các tình huống mà chúng tôi quan sát thấy đều là một điểm nghẽn.

Cho đến năm 2020, các câu hỏi xung quanh việc liệu việc sử dụng các phương pháp này có thể được áp dụng với hoạt động phân phối hay không đã được thúc đẩy bởi các nhóm ở nước ngoài, sự sẵn có về kỹ năng và cuối cùng là chi phí. Các nhà tích hợp hệ thống lớn đã dành hàng tỷ USD trong những năm 2000 để đầu tư vào các trung tâm phát triển gần bờ và ngoài khơi nên có thể hiểu được tại sao họ muốn làm mọi thứ có thể để cho phép các nhóm ở các trung tâm đó có thể sử dụng các phương pháp thực hành DevOps, tinh gọn và linh hoạt. Đối với Agilistas, điều này có thể gây khó chịu vì trọng tâm của Agile tập trung chủ yếu vào việc chuyển cuộc trò chuyện sang giá trị hơn là chi phí cũng như cách liên tục học hỏi và cải tiến để tối ưu hóa giá trị.

Năm 2020 chứng kiến một động lực mới và được nâng cao đáng kể cho các nhóm phân tán - đại dịch toàn cầu COVID-19. Chúng tôi đang ở giai đoạn đầu viết cuốn sách này khi đại dịch được công bố và các chính phủ bắt đầu nói về việc đóng cửa và các hạn chế nghiêm trọng khác. Từ tháng 3 năm 2020, hầu hết nếu không phải tất cả những người đọc cuốn sách này sẽ nhớ lại rằng cuộc sống cá nhân và công việc của họ đã thay đổi đáng kể. Phần lớn hoạt động phát triển và vận hành CNTT đột ngột được thực hiện tại nhà của nhân viên. Cơ hội cùng địa điểm cho bất kỳ hình thức cộng tác trực tiếp nào đã bị các công ty, chính phủ hạn chế nghiêm trọng và do rủi ro về sức khỏe, chính điều này đã gây ra sự gièm pha cho chính các cá nhân.

Giống như hầu hết, chúng tôi phải xoay vòng công việc của mình và sửa đổi cách làm việc của riêng mình với khách hàng trong thời kỳ đại dịch. Bất kể sở thích hay suy nghĩ trước đây về cách làm việc phân tán, giờ đây chúng tôi phải khám phá và đầu tư vào các phương pháp làm việc phân tán. Đối với cá nhân chúng tôi, điều này có nghĩa là ra mắt Khu dân cư ảo Phòng thí nghiệm đổi mới mở và các dịch vụ khác sẽ được cung cấp và phân phối từ xa.

Khi chúng tôi phát hành phần đầu tiên của cuốn sách này dưới dạng bản xem trước sớm để lấy ý kiến phản hồi, độc giả của chúng tôi đã rất khuyến khích chúng tôi khám phá việc sử dụng phân tán các thực tiễn nhiều hơn những gì chúng tôi dự định ban đầu. Vì vậy, chúng tôi sẽ làm như vậy. Trong mỗi phần của cuốn sách này, chúng tôi sẽ bao gồm một phần để chia sẻ kinh nghiệm áp dụng các phương pháp thực hành với các nhóm phân tán. Điều này sẽ bao gồm những câu chuyện và kinh nghiệm từ thực địa trong năm 2020 khi chúng tôi thực hiện việc này và nhiều bài học mà chúng tôi thu được từ việc đó. Nếu thích hợp, chúng tôi cũng sẽ bao gồm các chi tiết và liên kết đến các mẫu kỹ thuật số đã hỗ trợ chúng tôi.

Một lưu ý thận trọng. Chỉ đầu tư vào một công cụ như Mural hoặc Miro và có quyền truy cập vào nhiều mẫu sẽ không giúp bạn tiếp tục như bình thường với tất cả các phương pháp và kỹ thuật mà bạn có thể đã sử dụng trong phòng. Chúng tôi đã nêu tầm quan trọng của việc đạt được sự cân bằng giữa con người, quy trình và thay đổi công nghệ để có được Văn hóa và Thực hành DevOps thành công. Khi chuyển sang sử dụng chế độ phân tán - các nhóm cần tập trung nhiều hơn và liên tục vào con người, quy trình và công nghệ.

Đôi Lời Về Thế Giới 'Mở'

Thuật ngữ mở đã được sử dụng nhiều lần trong cuốn sách này và nó sẽ được sử dụng nhiều lần nữa. Chúng tôi làm việc cho một tổ chức mở, một công ty được xây dựng dựa trên các nguyên tắc và đặc điểm của nguồn mở. Chúng tôi đang sử dụng trải nghiệm của Phòng thí nghiệm Đổi mới Mở để kể nhiều câu chuyện của mình và tất cả các thực tiễn chúng tôi đang sử dụng đều được ghi lại và sẽ tiếp tục phát triển trong Thư viện Thực hành Mở.



THE WORLD OF
OPEN

Hình 3.7: Mặc định mở

Chúng tôi tin tưởng mạnh mẽ rằng văn hóa mở và các thực tiễn mở sử dụng công nghệ mở sẽ tạo nên loại cocktail tốt nhất để chuyển đổi thành công.

Phần kết luận

Trong chương này, chúng tôi đã giới thiệu PetBattle và cốt truyện của ứng dụng dành cho người yêu thích, sẽ hình thành nên nghiên cứu điển hình thú vị mà chúng tôi sẽ sử dụng trong suốt cuốn sách này.

Chúng tôi cũng giới thiệu cách chúng tôi thường xuyên chia sẻ những câu chuyện và ví dụ thực tế từ công việc chúng tôi đã thực hiện với khách hàng của mình.

Cuối cùng, chúng tôi trình bày các phần còn lại của cuốn sách và những nội dung chúng tôi sẽ khám phá trong từng phần đó.

Phần giới thiệu của chúng tôi đã hoàn tất. Hãy bắt đầu tìm hiểu Vòng lặp Mobius và khám phá các phương pháp được sử dụng nhiều nhất của chúng tôi. Trước khi đi vào vòng lặp, chúng ta sẽ đi tới phần móng mà vòng lặp sẽ đứng trên đó. Trong chương tiếp theo, chúng ta sẽ bắt đầu bằng việc xây dựng nền tảng vô cùng quan trọng của văn hóa.

Khen ngợi văn hóa DevOps và Thực hành với OpenShift

"Tạo ra các nhóm thành công, có hiệu suất cao không phải là điều dễ dàng. Văn hóa DevOps và Thực hành với OpenShift cung cấp hướng dẫn thực tế từng bước để giải phóng sức mạnh của các quy trình mở và công nghệ làm việc cùng nhau."

—Jim Whitehurst, Chủ tịch, IBM

"Cuốn sách này chứa đầy trí tuệ của Tim, Mike, Noel và Donal và được minh họa một cách đáng yêu của Ilaria. Mọi nguyên tắc và thực hành trong cuốn sách này đều được hỗ trợ bởi những câu chuyện tuyệt vời của những người đã tham gia vào hành trình học tập của họ. Các tác giả đam mê hình dung mọi thứ và mỗi chương đều chứa đầy những ví dụ trực quan mạnh mẽ. Luôn có nội dung nào đó dành cho mỗi độc giả và bạn sẽ thấy mình quay lại xem các ví dụ này nhiều lần." —Jeremy Brown, Giám đốc Công nghệ/Giám đốc Sản phẩm tại Traveledoo, một Công ty của Expedia

"Cuốn sách này mô tả rõ ý nghĩa của việc hợp tác với Phòng thí nghiệm đổi mới mở Red Hat, triển khai DevOps công nghiệp và đạt được sự linh hoạt trong kinh doanh bằng cách lắng nghe nhóm. Tôi đã trực tiếp trải nghiệm điều này. Sử dụng phương pháp được giải thích trong cuốn sách này, chúng tôi đã đạt được một cấp độ về sự hợp tác và gắn kết trong nhóm mà chúng tôi chưa từng trải qua trước đây, kết quả không mất nhiều thời gian và thành công là điều tất yếu mà tôi thấy là yếu tố thành công chính là sự thay đổi trong tư duy giữa các thành viên trong nhóm và trong cách tiếp cận này. đã giúp chúng tôi lái xe." —Michael Denecke, Trưởng phòng Công nghệ Thử nghiệm tại Volkswagen AG

"Cuốn sách này chứa đầy kinh nghiệm, niềm vui, niềm đam mê và cách thực hành tuyệt vời. Nó chứa tất cả các thành phần cần thiết để tạo ra văn hóa DevOps hiệu suất cao...thật tuyệt vời!" —John Faulkner-Willcocks, Trưởng phòng Văn hóa Huấn luyện và Giao hàng, JUST

"DevOps có cơ hội thay đổi cách các nhóm phần mềm làm việc và sản phẩm họ phân phối. Để thực hiện lời hứa này, chương trình DevOps của bạn phải bắt nguồn từ con người. Cuốn sách này giúp bạn khám phá những tư duy, nguyên tắc và thực tiễn sẽ thúc đẩy kết quả thực sự." —Douglas Ferguson, Người sáng lập Kiểm soát Điện áp, Tác giả cuốn Magical Meets and Beyond the Prototype

"Độc thú vị và mãnh liệt! Bằng cách nào đó, các tác giả đã gói gọn văn hóa Red Hat và sự thể hiện trong cuốn sách này."

—Jonas Frydal, Giám đốc Volvo Cars

"Cuốn sách này thực sự có giá trị đối với tôi. Tôi có thể liên kết từng đoạn tôi đọc với hành trình chúng tôi đã thực hiện trong thời gian làm việc với Phòng thí nghiệm đổi mới mở Red Hat. Đó là khoảng thời gian căng thẳng nhưng cũng bổ ích, học được rất nhiều về văn hóa, sự cởi mở, linh hoạt và cách sự kết hợp của chúng có thể giúp mang lại giá trị kinh doanh quan trọng trong một khoảng thời gian ngắn. Phát biểu từ kinh nghiệm cá nhân của tôi, chúng tôi đã hỗ trợ lẫn nhau, nhóm của tôi mang đến kiến thức sâu rộng trong ngành và nhóm của Red Hat mang đến những phương pháp hay cho kiến trúc dựa trên đám mây. Điều này giúp có thể phát minh lại cách thử nghiệm công nghệ điện tử trên xe đồng thời thúc đẩy OpenShift của Red Hat theo hướng DevOps công nghiệp. Tôi mong muốn giữ một bản cứng của cuốn sách tại bàn làm việc của mình để dễ dàng xem xét." —Marcus Greul, Giám đốc chương trình tại CARIAD, một công ty thuộc Tập đoàn Volkswagen

"Sự đổi mới đòi hỏi nhiều thứ hơn là ý tưởng và công nghệ. Nó cần mọi người được lãnh đạo tốt và các khái niệm cũng như hướng dẫn về 'Lãnh đạo mở' trong Thực hành và văn hóa DevOps với OpenShift cần được yêu cầu đọc đối với bất kỳ ai đang cố gắng đổi mới, trong mọi môi trường, với bất kỳ nhóm nào." —Patrick Heffernan, Giám đốc Thực hành và Nhà phân tích Chính, Technology Business Research Inc.

"Ồ! Đây hẳn phải là cuốn sách DevOps phi hư cấu hay nhất mà tôi từng đọc. Tôi không thể tin được nhóm đã nắm bắt được bản chất của nội dung chương trình nội trú của Phòng thí nghiệm đổi mới mở tốt đến mức nào. Sau khi đọc, bạn sẽ có một kiến thức vững chắc hộp công cụ gồm các nguyên tắc khác nhau và các biện pháp thực hành cụ thể để xây dựng văn hóa DevOps, nhóm và các quy trình lấy con người làm đầu để thay đổi cách bạn sử dụng công nghệ nhằm hoạt động như một hệ số nhân trong tổ chức của mình." —Antti Jaakkonen, Huấn luyện viên Lean Agile, DNA Plc

"Thú vị! Cuốn sách này là cuốn sách phải đọc dành cho tất cả các doanh nhân công nghệ muốn xây dựng hệ thống có khả năng mở rộng và các công ty bền vững. Thành công bây giờ đã được trao cho bạn."

—Jeep Kline, Nhà đầu tư mạo hiểm, Doanh nhân

"Trong nền kinh tế ưu tiên kỹ thuật số, nơi công nghệ được đưa vào mọi hoạt động kinh doanh, văn hóa đổi mới và DevOps là một phần không thể thiếu trong việc tạo ra các giá trị tổ chức mới và lợi thế cạnh tranh. Một hướng dẫn thực tế và dễ hiểu cho cả người thực hành công nghệ và lãnh đạo doanh nghiệp sẽ rất hữu ích đối với các công ty đẩy nhanh các chiến lược Chuyển đổi kỹ thuật số (DX) của họ để phát triển mạnh mẽ trong một thế giới đã thay đổi." —Sandra Ng, Phó Chủ tịch Nhóm, Thực hành CNTT

"Văn hóa và thực hành DevOps với OpenShift là sự chất lọc nhiều năm kinh nghiệm thành một nguồn tài nguyên tuyệt vời có thể được sử dụng làm sách công thức cho các nhóm khi họ hình thành và phát triển hoặc làm hướng dẫn tham khảo cho các nhóm trưởng thành khi họ tiếp tục phát triển." —David Worthington, Huấn luyện viên chuyển đổi linh hoạt, Ngân hàng DBS, Singapore

Nhận bản sao của riêng bạn

Mua sách bìa mềm từ Packt Publishing
để tham khảo nhanh chóng và dễ dàng.



Văn hóa và thực hành DevOps với OpenShift

Bản quyền © 2021 Nhà xuất bản Packt

Mọi quyền được bảo lưu. Không phần nào của cuốn sách này được phép sao chép, lưu trữ trong hệ thống truy xuất hoặc truyền dưới bất kỳ hình thức nào hoặc bằng bất kỳ phương tiện nào mà không có sự cho phép trước bằng văn bản của nhà xuất bản, ngoại trừ trường hợp trích dẫn ngắn gọn được nhúng trong các bài báo hoặc bài phê bình quan trọng.

Mọi nỗ lực đã được thực hiện trong quá trình biên soạn cuốn sách này để đảm bảo tính chính xác của thông tin được trình bày. Tuy nhiên, thông tin trong cuốn sách này được bán mà không có bảo đảm, dù rõ ràng hay ngụ ý. (Các) tác giả cũng như Nhà xuất bản Packt hoặc các đại lý và nhà phân phối của nó sẽ không chịu trách nhiệm pháp lý về bất kỳ thiệt hại nào gây ra hoặc bị cáo buộc là do cuốn sách này gây ra trực tiếp hoặc gián tiếp.

Nhà xuất bản Packt đã nỗ lực cung cấp thông tin nhãn hiệu về tất cả các công ty và sản phẩm được đề cập trong cuốn sách này bằng cách sử dụng vốn thích hợp. Tuy nhiên, Packt Publishing không thể đảm bảo tính chính xác của thông tin này.

Tác giả: Tim Beattie, Mike Hepburn, Noel O'Connor và Donal Spring
Người minh họa: Ilaria Doria

Người đánh giá kỹ thuật: Ben Silverman
Biên tập viên quản lý: Aditya Datar và Siddhant Jain
Biên tập viên mua lại: Ben Renow-Clarke

Biên tập sản xuất: Deepak Chavan
Ban biên tập: Vishal Bodwani, Ben Renow-Clarke, Edward Doxey, Alex Patterson, Arijit Sarkar, Jake Smith và Lucy Wan

Xuất bản lần đầu: Tháng 7 năm
2021 Số tham chiếu sản xuất:
1100821 ISBN: 978-1-80020-236-8

Được xuất bản bởi Packt Publishing Ltd.
Livery Place, 35 Livery Street,
Birmingham, B3 2PB, Vương quốc Anh.

www.packt.com