

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

-- -- oOo-- --



BÁO CÁO THỰC TẬP THỰC TẾ
HỆ THỐNG THÔNG TIN (CT451)

Cơ quan thực tập: TMA Solutions

Thời gian thực tập: 21/05/2018 - 13/07/2018

Cán bộ hướng dẫn: Anh. Trần Duy Long

Giảng viên hướng dẫn: ThS. Nguyễn Cao Hồng Ngọc

Sinh viên thực hiện: Nguyễn Việt Thảo Nguyễn B1400833

TP.Hồ Chí Minh, ngày 16/07/2018

LỜI CẢM ƠN

Thực tập thực tế là một học phần rất quan trọng và cần thiết đối với các bạn sinh viên. Bản thân em thật sự rất may mắn khi có cơ hội trở thành thực tập sinh cũng như là nhân viên thử việc tại TMA Solutions. Mặc dù thời gian thực tập tại công ty là rất ngắn nhưng nhờ có sự chỉ bảo tận tình của các anh chị mà em đã học hỏi, tiếp cận được với những công nghệ mới cùng kỹ thuật hiện đại.

Em xin chân thành cảm ơn các Thầy Cô của Khoa CNTT & TT đã truyền đạt những kiến thức bổ ích cũng như đã đến thăm, lắng nghe những tâm tư và hướng dẫn em trong thời gian thực tập tại công ty.

Trong quá trình thực tập tại công ty, em khó tránh khỏi các sai sót, rất mong Thầy, Cô và các anh chị bỏ qua. Bên cạnh đó, do kiến thức còn hạn hẹp nên bài báo cáo cũng khó tránh khỏi những thiếu sót. Rất mong nhận được ý kiến đóng góp từ Cô và các anh chị trong công ty.

Cuối lời, em kính chúc Cô nhiều sức khỏe và công tác tốt. Kính chúc các anh chị trong công ty TMA Solutions luôn dồi dào sức khỏe, đạt được nhiều thành công trong công việc và cuộc sống.

Em xin chân thành cảm ơn!

PHIẾU ĐÁNH GIÁ BÁO CÁO KẾT QUẢ THỰC TẬP M-TT-04
HỌC KỲ 3 – 2017-2018
(Dùng cho giáo viên chấm báo cáo thực tập)

Họ và tên cán bộ chấm báo cáo:..... Nguyễn Cao Hồng Ngọc

Họ tên sinh viên thực tập:..... Nguyễn Việt Thảo NguyênMã số SV: B1400833

Nội dung đánh giá	Điểm tối đa	Điểm chấm
I. Hình thức trình bày	1.0	
I.1 Đúng format của khoa (Trang bìa, trang lời cảm ơn, trang đánh giá thực tập của khoa, trang mục lục và các nội dung báo cáo). Sử dụng đúng mã và font tiếng Việt (Unicode Times New Roman, Size 13)	0.5	
I.2 Trình bày mạch lạc, súc tích, không có lỗi chính tả	0.5	
II. Phiếu theo dõi	4.75	
II.1 Có lịch làm việc đầy đủ cho 8 tuần	0.25	
II.2 Số buổi thực tập tại cơ quan trong 1 tuần ≥ 6 ; ít hơn 6 buổi 0.0 điểm	1.0	
II.3 Hoàn thành tốt kế hoạch công tác ghi trong lịch làm việc. Cách tính điểm = (điểm cộng của cán bộ hướng dẫn/100) x 3.5	3.5	
III. Nội dung thực tập (quyển báo cáo)	4.25	
III.1 Có được sự hiểu biết tốt về cơ quan nơi thực tập	0.5	
III.2 Phương pháp thực hiện phù hợp với nội dung công việc được giao	1.0	
III.3 Kết quả củng cố lý thuyết	0.5	
III.4 Kết quả rèn luyện kỹ năng thực hành	0.5	
III.5 Kinh nghiệm thực tiễn thu nhận được	0.5	
III.6 Kết quả công việc có đóng góp cho cơ quan nơi thực tập	1.25	
TỔNG CỘNG	10.0	
Điểm trừ		
Điểm còn lại		

Lưu ý:

Không dự họp để nghe phổ biến TTTT: trừ 1 điểm

Không gửi phiếu giao việc về khoa đúng hạn (đến 17.05.2018 theo dấu bưu điện): trừ 1 điểm

....., ngày....tháng 07 năm 2018

GIÁO VIÊN CHĂM BÁO CÁO

(ký tên)

Nguyễn Cao Hồng Ngọc

MỤC LỤC

LỜI CẢM ƠN	I
PHIẾU ĐÁNH GIÁ BÁO CÁO KẾT QUẢ THỰC TẬP	II
CHƯƠNG 1. GIỚI THIỆU SƠ LƯỢC VỀ CÔNG TY TMA SOLUTIONS.....	1
1. TMA – Hình thành và phát triển.....	1
2. Sơ đồ tổ chức TMA Solutions	1
CHƯƠNG 2. CÁC HOẠT ĐỘNG CHUYÊN NGÀNH VÀ MÔI TRƯỜNG LÀM VIỆC TẠI TMA SOLUTIONS	2
1. Các hoạt động chuyên ngành.....	2
2. Môi trường làm việc tại TMA Solutions	2
CHƯƠNG 3. NỘI DUNG CÔNG VIỆC ĐƯỢC PHÂN CÔNG VÀ PHƯƠNG PHÁP THỰC HIỆN.....	3
1. Công việc được phân công	3
2. Phương pháp thực hiện.	3
CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC.....	5
1. Tổng quan về Test Automation	5
1.1. Test Automation là gì.	5
1.2. Tại sao cần Test Automation	5
1.3. Khi nào cần áp dụng Test Automation.....	5
2. Các phương pháp kiểm thử tự động.	5
2.1. Ghi/ Phát lại (Record/ Playback)	5
2.2. Phương pháp hướng dữ liệu.....	6
3. Cơ bản về Robot Framework.....	6
3.1. Giới thiệu về Robot Framework.	6
3.2. Robot Framework Test Cases	7

4.	Tích hợp liên tục (Continuous Integration).....	7
4.1.	CI – Tích hợp liên tục là gì?	7
4.2.	Tại sao nên dùng CI?	8
4.3.	Workflow của CI.	8
4.4.	Jenkins.	9
4.4.1.	Blue Ocean.	9
4.4.2.	Jenkins Pipeline	9
4.4.3.	Bitbucket.....	10
4.4.4.	Webhook.....	10
5.	Kinh nghiệm thực tiễn đã tích lũy được.	11
6.	Chi tiết kết quả công việc đã đóng góp cho công ty.	12

CHƯƠNG 1. GIỚI THIỆU SƠ LƯỢC VỀ CÔNG TY TMA SOLUTIONS

1. TMA – Hình thành và phát triển

- TMA Solutions(gọi tắt là TMA) : Công ty TNHH Giải pháp phần mềm Tường Minh là một công ty tư nhân, 100% vốn Việt Nam, kinh doanh các dịch vụ liên quan đến phát triển phần mềm.
- Được thành lập vào 03/1997 bởi bà Bùi Ngọc Anh, lúc bấy giờ công ty chỉ mới có 6 kỹ sư. Sau hơn 20 năm thành lập và phát triển vững mạnh, số lượng kỹ sư CNTT đang làm việc tại công ty hiện nay đã hơn 2000 nhân viên.
- Hiện nay TMA đang có 6 lab tại TP.HCM và 1 lab tại Bình Định đã đi vào hoạt động từ ngày 11/6/2018. Ngoài ra TMA còn có 5 chi nhánh ở nước ngoài: Nhật Bản, Canada, Mỹ, Châu Âu và Úc
- TMA hiện đang là công ty phần mềm hàng đầu Đông Nam Á và hiện là đối tác tin cậy có nhiều năm kinh nghiệm thực hiện các dự án lớn & phức tạp với hơn 100 khách hàng đến từ hơn 25 quốc gia trên toàn thế giới.
- Lĩnh vực hoạt động của TMA hiện nay là:
 - Tích hợp hệ thống.
 - Xuất khẩu phần mềm.
 - Giải pháp phần mềm.
 - Cung cấp các giải pháp dịch vụ viễn thông.
 - Đào tạo sinh viên và nhân viên về các kiến thức phần mềm và kỹ năng mềm.

2. Sơ đồ tổ chức TMA Solutions

- TMA Solutions:
 - 4 DG được chia thành 17 DC.
 - Hội đồng quản trị (BOD).
 - Business & Shared Services.
 - CDF.
 - Delivery Shared Services.
 - Tài chính và Pháp lý.
 - HR/ Admin Support.
 - TMA Overseas.

CHƯƠNG 2. CÁC HOẠT ĐỘNG CHUYÊN NGÀNH VÀ MÔI TRƯỜNG LÀM VIỆC TẠI TMA SOLUTIONS

1. Các hoạt động chuyên ngành.

- Phát triển phần mềm.
- Kiểm thử phần mềm.
- Porting & Chuyển đổi.
- Hỗ trợ sản xuất.
- Dịch vụ quản lý CNTT.
- Thiết kế trực quan.

Cung cấp giải pháp cho nhiều lĩnh vực:

- Viễn thông.
- Tài chính và bảo hiểm.
- Thương mại điện tử, bán lẻ và phân phối.
- Nhà cung cấp phần mềm độc lập.
- Phương tiện truyền thông và xuất bản.
- Chăm sóc sức khỏe.
- Giáo dục.
- Khách sạn và du lịch.
- Hậu cần và giao thông.

2. Môi trường làm việc tại TMA Solutions

- Thời gian làm việc: Từ thứ 2 đến thứ 6, buổi sáng từ 8h30 đến 12h, chiều từ 13h30 đến 18h.
- Môi trường làm việc tại TMA thân thiện và chuyên nghiệp không chỉ ở cơ sở vật chất mà còn thể hiện qua chính sách hỗ trợ tài chính.
- Ngay khi gia nhập vào TMA, các bạn sinh viên đều được giới thiệu cụ thể về công ty, các quy định cũng như vị trí làm việc. Trong quá trình thực tập kết hợp thử việc, em được training về kỹ năng làm việc, tham gia CLB Toastmaster để bổ sung kỹ năng tiếng Anh cần thiết khi làm việc tại TMA.

CHƯƠNG 3. NỘI DUNG CÔNG VIỆC ĐƯỢC PHÂN CÔNG VÀ PHƯƠNG PHÁP THỰC HIỆN.

1. Công việc được phân công

Tuần	Thời gian	Nội dung công việc
Tuần 1	21/05/2018 - 22/05/2018	Cài đặt PC: các phần mềm và công cụ sẽ sử dụng.
	23/05/2018 - 25/05/2018	Hỗ trợ test tính năng mới của dự án Omena.
Tuần 2	28/05/2018 - 30/05/2018	Tìm hiểu Java, các phần mềm và công cụ liên quan.
	31/05/2018 - 01/06/2018	Cài đặt môi trường Codelab trên local
Tuần 3	04/06/2018 - 05/06/2018	Linux, giao diện dòng lệnh và các công cụ liên quan.
	06/06/2018 - 08/06/2018	Test tính năng Omena.
Tuần 4	11/06/2018 - 12/06/2018	Tìm hiểu tổng quan về automation test, các framework và công cụ hỗ trợ.
	13/06/2018 - 15/06/2018	Thực hiện Automation test
Tuần 5	18/06/2018 - 22/06/2018	Cài đặt môi trường Automation
Tuần 6, 7	25/06/2018 - 06/07/2018	Viết automation test cho dự án Codelab, thực hiện chạy trên local
Tuần 8	09/07/2018 - 13/07/2018	Automation test và Tích hợp liên tục và áp dụng lên Codelab

2. Những việc đã làm được

- Tìm hiểu về Test Automation và các framework hỗ trợ hiện nay.
- Tìm hiểu về Java (được sử dụng để viết Automation Test case).
- Tìm hiểu về Robotframework và kết hợp với ngôn ngữ Java.
- Xây dựng Automation Platform cho hệ thống Codelab sử dụng Robotframework.
- Automation test và Tích hợp liên tục và áp dụng lên Codelab
- Hoàn thiện chức năng và demo sản phẩm.

3. Phương pháp thực hiện.

- Nghiên cứu các tài liệu liên quan đến Test Automation.

- Cài đặt Eclipse.
- Cấu hình Maven.
- Cài đặt Python 2.7.15
- Cài đặt wxPython 4.0.3
- Cài đặt PIP (Python Package Manager) 10.0.1
- Cài đặt Testing framework: Robot Framework 3.0.4
 - Framework dùng để kiểm thử này cung cấp nền tảng kiểm thử cho tester dựa trên ngôn ngữ lập trình Python. Để tiếp cận nền tảng kiểm thử này, tester cần viết kịch bản kiểm thử theo hướng từ khóa (Keyword driven) và hướng dữ liệu (data driven).
 - Robot Framework đưa ra kết quả thực thi các kịch bản kiểm thử và các log ở dạng html, giúp chúng ta đọc và phân tích kết quả dễ dàng hơn.
 - Robot Framework có hỗ trợ chức năng đánh dấu các kịch bản kiểm thử, cho phép chúng ta lựa chọn chạy bất kỳ kịch bản nào một cách tiện lợi và nhanh chóng.
 - Thế mạnh lớn nhất của Robot Framework là khả năng chạy trên nhiều hệ điều hành khác nhau mà không cần chỉnh sửa kịch bản hay các từ khóa ở tầng dưới.
- Cài đặt thư viện hỗ trợ Robot framework: Selenium2Library.
 - Selenium2Library được sử dụng để kiểm thử trên nền Web, được fork từ SeleniumLibrary và được bổ sung để sử dụng Selenium 2 và WebDriver.
 - Selenium2Library là ngôn ngữ sát với ngôn ngữ thực tế của người dùng, bạn mong muốn action gì thì chỉ cần gõ từ khóa tương ứng.
- Cài đặt RIDE (Standalone RobotFramework Test Data Editor)
 - RIDE là một IDE để xây dựng kiểm thử bằng cách sử dụng Robot Framework.
 - Vì RIDE được phát triển bằng cách sử dụng wxPython nên chúng ta cần cài bộ Tool wxPython 4.0.3 Có hỗ trợ Unicode để chạy RIDE.
- Cài đặt Jenkins 2.129

CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC.

1. Tổng quan về Test Automation

1.1. Test Automation là gì.

Test Automation là việc sử dụng phần mềm để kiểm soát việc thực hiện kiểm thử, so sánh kết quả thực tế và kết quả mong đợi, thiết lập các điều kiện tiên quyết cho việc kiểm thử và chức năng báo cáo kiểm thử.

1.2. Tại sao cần Test Automation

Test Automation cho phép thực hiện kiểm thử một cách hiệu quả hơn rất nhiều so với việc kiểm thử thủ công.

- Chạy kiểm thử hồi quy trên phiên bản mới của chương trình.
- Thực hiện kiểm thử thủ công sẽ khó hoặc không khả thi.
- Tính nhất quán và lặp lại của kiểm thử.
- Sử dụng tài nguyên tốt hơn.
- Sớm đưa ra thị trường.

1.3. Khi nào cần áp dụng Test Automation.

Kiểm thử tự động có thể thực hiện nhằm mang lại hiệu quả về chi phí nếu một số common sense sau đây được áp dụng cho quy trình:

- Chỉ việc kiểm thử tự động được lặp lại.
- Chọn một công cụ kiểm thử phù hợp nhất với yêu cầu kiểm thử của tổ chức hoặc công ty của bạn.
- *Tránh sử dụng phương pháp “Ghi/ Phát lại”.*
- *Áp dụng phương pháp kiểm thử tự động theo hướng dữ liệu.*

2. Các phương pháp kiểm thử tự động.

2.1. Ghi/ Phát lại (Record/ Playback)

Record/ Playback là một tính năng của hầu hết các công cụ kiểm thử cho phép người dùng ghi lại các tổ hợp phím và di chuyển chuột trong quá trình tương tác với AUT (Application Under Test).

Các tổ hợp phím và di chuyển chuột được ghi lại như một kịch bản, và sau đó có thể “phát lại” trong quá trình thực thi kiểm thử.

- Lợi ích của việc sử dụng dụng Record/ Playback.
 - Hữu dụng cho việc xác định công cụ đang cố gắng xử lý và tương tác với AUT như thế nào.
 - Đưa ra cho bạn một số ý tưởng về việc phát triển kịch bản kiểm thử như thế nào.
- Bất lợi của việc sử dụng Record/ Playback
 - Kịch bản sẽ bao gồm các giá trị được mã hóa cứng (không thể thay đổi).
 - Chi phí cao cho việc bảo trì kịch bản.
 - Kịch bản không đáng tin tưởng.
 - Test cần phải ghi lại nếu: tester gây ra lỗi trong việc nhập dữ liệu hoặc có sự thay đổi của AUT.

2.2. Phương pháp hướng dữ liệu.

Kiểm thử hướng dữ liệu là việc tự động tiếp cận mà việc điều hướng và chức năng của kịch bản test được chỉ dẫn thông qua dữ liệu bên ngoài.

Framework kiểm thử hướng dữ liệu được thiết kế để cho phép việc thực thi của một lượng lớn các biến thể chức năng của dữ liệu kiểm thử bằng cách sử dụng một lượng nhỏ của kịch bản kiểm thử và chức năng.

3. Cơ bản về Robot Framework.

3.1. Giới thiệu về Robot Framework.

- Cho phép dễ sử dụng cú pháp bảng để tạo test cases theo cách đồng nhất.
- Cung cấp khả năng để tạo những keyword cấp cao có thể tái sử dụng từ những keyword đã có sẵn.
- Cung cấp những báo cáo kết quả (report) và nhật ký (log) ở dạng HTML.
- Là một nền tảng và ứng dụng độc lập.
- Cung cấp thư viện API đơn giản cho việc tạo các thư viện test tùy chỉnh mà có thể thực hiện nguyên bản với cả Python và Java.
- Cung cấp hỗ trợ cho Selenium cho việc test web, Java GUI test, ..
- Hỗ trợ tạo test cases theo hướng dữ liệu.



- Cung cấp tag để phân loại và lựa chọn test cases để thực thi.

3.2. Robot Framework Test Cases

- Test suites
 - Là bộ sưu tập những test case trong Robot Framework.
 - Test suites có thể là file hoặc thư mục.
- Keywords: Test cases được tạo nên bởi Keywords có thể đến từ 2 nguồn
 - Keyword của thư viện: Keyword cấp thấp.
 - Keyword do người dùng xác định: Keyword cấp cao.
- Libraries: thường được import bằng Library setting, đôi khi cũng có thể import bằng Library Keyword.
- Variables: Robot Framework có hai loại variable.
 - Scalar Variables: khi sử dụng Scalar variables trong test dữ liệu, nó sẽ thay thế những giá trị mà nó chỉ định tới.
 - List Variables: là biến phức có thể có nhiều giá trị chỉ định tới nó.
- Setups and Teardowns:
 - Suite Setup and Suite Teardown: Nếu muốn điều kiện nhất định thực thi trước hoặc sau mỗi test suite.
 - Test Setup and Test Teardown: Nếu muốn điều kiện nhất định thực thi trước hoặc sau mỗi lần test.
 - Custom Setup and Custom Teardown: Nếu muốn điều kiện nhất định thực thi cho từng test.
- Tags: giúp việc chạy test hiệu quả hơn, có thể test cụ thể chứ không cần lặp lại toàn bộ quy trình.
- Resource File: User keywords và các giá trị trong file test case và các file khởi tạo test suite chỉ có thể được sử dụng trong files mà nó được tạo nên, còn resource files cung cấp một cơ chế để chia sẻ chúng.

4. Tích hợp liên tục (Continuous Integration)

4.1. CI – Tích hợp liên tục là gì?

Tích hợp liên tục (CI) là phương pháp phát triển phần mềm đòi hỏi các thành viên trong nhóm tích hợp công việc thường xuyên vào một kho lưu trữ. Sau đó, mỗi tích hợp đều

được xác minh bằng việc tự động build và tự động test. Phương pháp tiếp cận này giúp giảm bớt vấn đề về tích hợp và cho phép phát triển phần mềm gắn kết nhanh hơn.

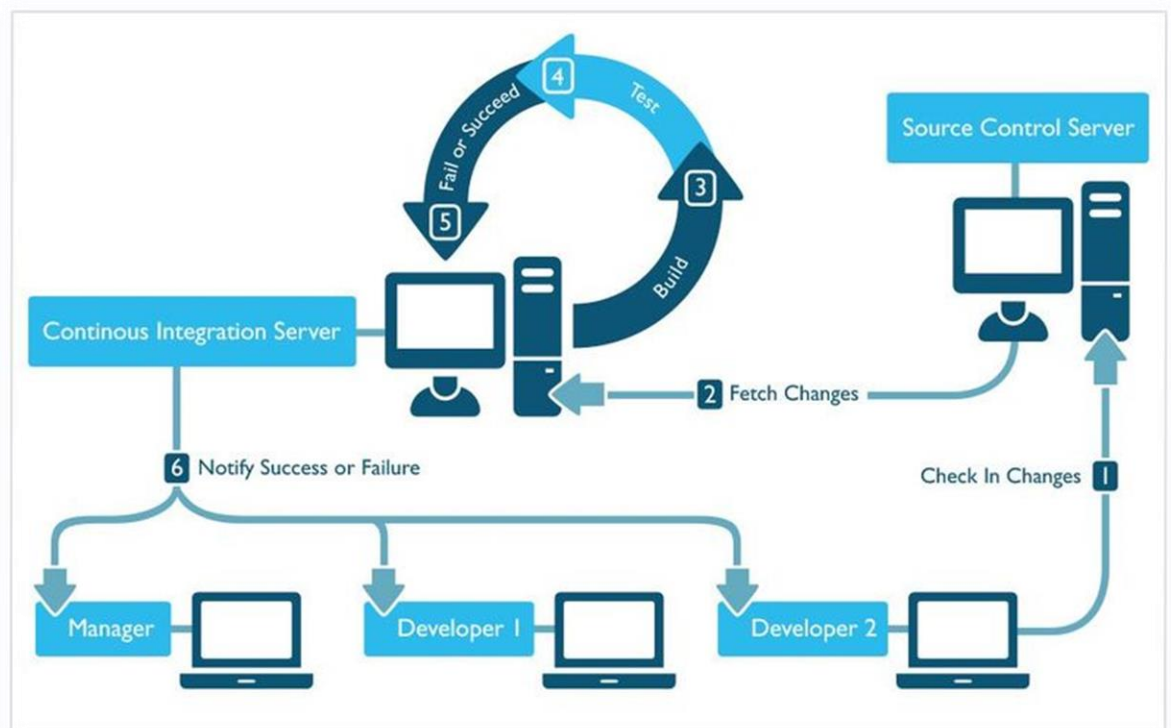
4.2. Tại sao nên dùng CI?

CI là phương pháp mà nhiều team Agile lựa chọn sử dụng nhằm đảm bảo code của project luôn build được, luôn pass toàn bộ các test case. Các công đoạn test sẽ được hệ thống CI server thực hiện tự động giúp ta dễ dàng biết được tình trạng của một branch, một commit nào đó mà không cần lấy source về test thử. Do đó tốc độ phát triển được tăng lên.

Một số framework được sử dụng phổ biến hiện nay như: TFS, TeamCity, Hudson, Jenkins, Travis CI, GitLab CI,...

4.3. Workflow của CI.

Workflow của CI được chia thành 6 bước.



- Bước 1: Các developer thực hiện các ticket và đẩy code lên thư mục nơi sẽ lưu giữ code của Project.
- Bước 2: CI server lắng nghe và thực hiện kéo code về, build rồi test code.
- Bước 3,4,5: CI Server sẽ build, test, cập nhật là trạng thái build của bản source đó.

- Bước 6: CI Server sẽ thông báo đến người quản lý và nhóm phát triển về tình trạng của bản code mới. Trong trường hợp build thất bại, các thành viên trong team sẽ được thông báo và developer có trách nhiệm phải sửa lỗi và đẩy code sửa lỗi đó lên repositories.

4.4. Jenkins.

Jenkins là một phần mềm mã nguồn mở dạng server, được viết bằng ngôn ngữ Java, chạy trên nền server với sự hỗ trợ của Apache Tomcat. Jenkins có thể được sử dụng để tự hóa tất cả các loại nhiệm vụ liên quan đến xây dựng, thử nghiệm và phân phối hoặc triển khai phần mềm.



Việc kích hoạt build dự án bằng Jenkins có thể được thực hiện bằng nhiều cách: dựa theo các lần commit trên mã nguồn, theo các khoảng thời gian, kích hoạt qua URL, kích hoạt sau khi các job khác được build,... Các chức năng của Jenkins có thể được mở rộng thông qua các plugin.

4.4.1. Blue Ocean.

Blue Ocean – một giao diện người dùng (UI) mà bất kỳ thành viên nào cũng có thể sử dụng Jenkins mà không cần phải có sự tinh thông về nó.

Các chức năng chính của Blue Ocean:

- Các hình ảnh tinh vi về Pipelines phân phối liên tục (CD- continuous delivery), cho phép hiểu nhanh và trực quan về trạng thái Pipeline của bạn.
- Trình biên tập Pipeline – giúp người dùng dễ dàng tiếp cận Pipeline thông qua quá trình tạo Pipeline một cách trực quan.
- Cá nhân hóa để phù hợp với nhu cầu của từng thành viên trong team.
- Xác định chính xác khi nào can thiệp là cần thiết. Blue Ocean sẽ cho biết cần chú ý đến Pipeline nào, tạo thuận lợi cho việc xử lý ngoại lệ và tăng năng suất.
- Tích hợp tự nhiên cho branch và pull request, cho phép năng suất phát triển tối đa khi cộng tác ở Github và Bitbucket.

4.4.2. Jenkins Pipeline

Jenkins Pipeline là một bộ các plugins hỗ trợ thực hiện và tích hợp liên tục phân phối các pipeline vào Jenkins. Pipeline cung cấp khả năng mở rộng 1 bộ các công cụ từ

đơn giản đến phức tạp cho việc chuyển các pipeline thông qua Pipeline domain-specific language (DSL) syntax.

4.4.3. Bitbucket

Bitbucket là dịch vụ web quản lý phiên bản phân tán cho Git, được viết bằng ngôn ngữ lập trình Python và sử dụng Django web framework. Với Bitbucket và Git thành viên có thể quản lý phiên bản (mã nguồn) của dự án của mình, từ đó team leader có thể theo dõi và đánh giá tiến độ công việc.



4.4.4. Webhook

Webhook cung cấp cách để cấu hình Bitbucket Cloud để thực hiện các yêu cầu tới máy chủ của bạn (hoặc một số dịch vụ bên ngoài khác) bất cứ khi nào có sự kiện xảy ra trong Bitbucket Cloud.

Một webhook bao gồm:

- Một chủ đề (subject) – tài nguyên tạo sự kiện. Hiện tại, nguồn tài nguyên là kho lưu trữ (repository) mà bạn tạo webhook.
- Một hoặc nhiều sự kiện (events) – sự kiện mặc định là việc push lên repository, nhưng bạn có thể chọn nhiều sự kiện khác mà có thể trigger đến webhook.
- Một URL – đích đến cuối cùng nơi mà bạn muốn Bitbucket gửi tải trọng sự kiện khi mà có một sự kiện phù hợp xảy ra.

Sử dụng webhook để tích hợp ứng dụng với Bitbucket Cloud, ví dụ như:

- Mỗi khi có một user nào đó push commit lên repository, bạn sẽ nhận được thông báo từ CI server của bạn để bắt đầu build.
- Mỗi khi có một user nào đó push commit hoặc tạo một pull request, bạn có thể sẽ muốn hiển thị thông báo trong ứng dụng của bạn.

Nếu không có webhook, nếu bạn muốn phát hiện khi nào sự kiện diễn ra trên Bitbucket Cloud, bạn cần phải thăm dò API. Tuy nhiên, thăm dò API là việc rất bất tiện, không hiệu quả và lại dễ xuất hiện lỗi. Hãy xem cách mà tin nhắn SMS hoạt động trên điện thoại, bạn không cần phải check tin nhắn mỗi 5 phút 1 lần để xem liệu

bạn có tin nhắn nào không bởi vì điện thoại sẽ gửi cho bạn thông báo. Tương tự, webhook hoạt động như một thông báo vì vậy API không cần phải check lại mỗi phút.

5. Kinh nghiệm thực tiễn đã tích lũy được.

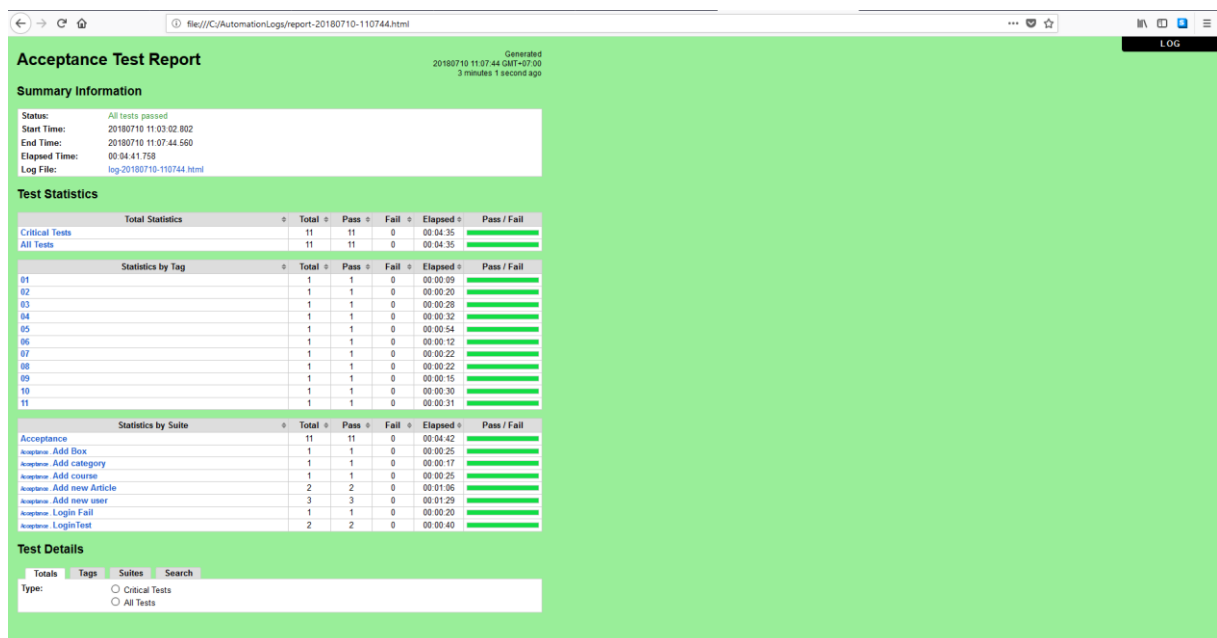
Trong quá trình thực tập tại công ty, bản thân em đã tích lũy được nhiều kinh nghiệm:

- Được tham gia dự án của công ty, được trải nghiệm trong một môi trường làm việc thực tế, theo quy trình bài bản, báo cáo tiến độ mỗi ngày, họp với team, họp trực tiếp với khách hàng, mọi công việc đều có deadline.
- Các kiến thức học ở trường đều là các kiến thức nền cơ bản rất quan trọng, không nắm được sẽ rất khó để bắt đầu cho các công nghệ mới được viết dựa trên nó.
- Công việc tại công ty đòi hỏi khá nhiều kiến thức mới, nằm ngoài chương trình đào tạo. Nên các bạn sinh viên ngay khi còn được đào tạo tại trường nên tranh thủ thời gian rảnh để tìm hiểu những công nghệ mới, những kiến thức mà các công ty phần mềm đang sử dụng để tránh bỡ ngỡ.
- Kỹ năng mềm là một nhân tố rất quan trọng khi thực tập tại công ty. Kỹ năng lắng nghe, giao tiếp, ứng xử, làm việc nhóm, giải quyết vấn đề sẽ được cải thiện trong quá trình làm việc.
- Học hỏi từ ngay các anh chị đi trước, người hướng dẫn - đã có kinh nghiệm làm việc tại công ty. Là người có nhiều năm kinh nghiệm trong ngành, họ sẽ giúp chúng ta tránh khỏi những sai sót mà người mới bắt đầu dễ vấp phải.
- Tự học là một kỹ năng rất cần thiết trong quá trình thực tập. Chúng ta cần rèn luyện tính tự học, tự tìm hiểu cách giải quyết vấn đề
- Trong 2 tháng thực tập tại công ty, em được tham gia vào **CLB Toastmaster**, giúp trau dồi kỹ năng nói tiếng Anh, mở rộng kiến thức ở nhiều lĩnh vực khác nhau. Ngoài ra còn có các buổi chuyên đề **Knowledge Sharing** nhằm chia sẻ các kiến thức chuyên môn nâng cao giữa các thành viên trong team.
- Nếu thể hiện tốt trong buổi phỏng vấn trước đó, hoặc trong quá trình thực tập hoàn thành công việc và có thái độ tốt thì các bạn sinh viên còn có cơ hội thử việc kết hợp thực tập. Đây là cơ hội tốt giúp bạn có nhiều khả năng trở thành nhân viên chính thức của công ty.

6. Chi tiết kết quả công việc đã đóng góp cho công ty.

Trong quá trình làm việc tại công ty TMA, em đã có cơ hội tham gia vào vào 2 dự án: **Omena và Codelab**.

- Vai trò của em ở dự án Omena là test manual cho các tính năng vừa được phát triển. Nhằm đảm bảo các tính năng phải vừa đáp ứng yêu cầu từ phía khách hàng, vừa mang tính hợp lý và tiện dụng cho người dùng. Trong quá trình test, **4 bugs** đã được tìm thấy, đã được báo cáo với team dev để fix bugs. Nhờ đó, sản phẩm đã được hoàn thiện hơn trước khi giao cho khách hàng và tránh những complain không đáng có.
- Ở dự án Codelab em tiến hành Test Automation. Cài đặt môi trường test, viết các test cases và thực hiện test trên web. Sau đó cài đặt Jenkins nhằm tích hợp liên tục, pull code về và thực hiện test trên local. Trong quá trình test, **10 bugs** được tìm thấy, được báo cáo ngay với team dev thông qua YouTrack để fix kịp thời.
 - Report của Automation test Codelab dưới dạng HTML



- Log của Automation test Codelab dưới dạng HTML

Acceptance Test Log

Generated: 20180710 11:07:44 GMT+07:00
50 seconds ago

REPORT

Test Statistics

Total Statistics		Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	11	11	0	00:04:35		
All Tests	11	11	0	00:04:35		

Statistics by Tag		Total	Pass	Fail	Elapsed	Pass / Fail
01		1	1	0	00:00:29	
02		1	1	0	00:00:29	
03		1	1	0	00:00:28	
04		1	1	0	00:00:32	
05		1	1	0	00:00:54	
06		1	1	0	00:00:12	
07		1	1	0	00:00:22	
08		1	1	0	00:00:22	
09		1	1	0	00:00:15	
10		1	1	0	00:00:30	
11		1	1	0	00:00:31	

Statistics by Suite		Total	Pass	Fail	Elapsed	Pass / Fail
Acceptance		11	11	0	00:04:42	
Acceptance: Add Box		1	1	0	00:00:25	
Acceptance: Add category		1	1	0	00:00:17	
Acceptance: Add course		1	1	0	00:00:25	
Acceptance: Add new Article		2	2	0	00:01:06	
Acceptance: Add new user		3	3	0	00:01:29	
Acceptance: Login Fail		1	1	0	00:00:29	
Acceptance: LoginTest		2	2	0	00:00:40	

Test Execution Log

Acceptance		00:04:41.758
Full Name:	Acceptance	
Source:	D:\Codelabs\master\codelab\test\testframework\acceptance	
Start / End / Elapsed:	20180710 11:03:02.302 / 20180710 11:07:44.588 / 00:04:41.758	
Status:	11 critical test, 11 passed, 0 failed 11 test total, 11 passed, 0 failed	
→ Add Box		00:00:24.672
→ Add category		00:00:17.440
→ Add course		00:00:24.533
→ Add new Article		00:01:06.382

Vì tính bảo mật của dự án nói riêng cũng như công ty nói chung nên không thể trình bày chi tiết nội dung dự án ở đây. Rất mong quý Thầy/ Cô thông cảm.