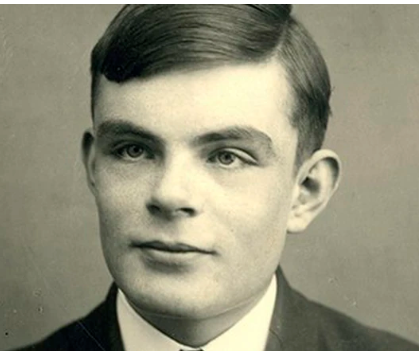
**Đề tài**

**Kiến trúc RESTful API và ứng dụng trong hệ thống IoT**

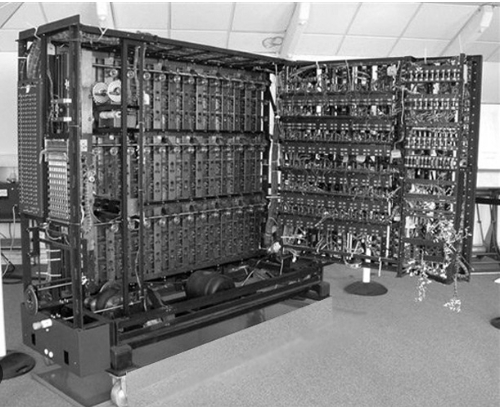
**Chương 1 : Tổng quan về phát triển phần mềm**

* 1. **Lịch sử của phần mềm (khoảng 5 trang)**
* Trong thế giới ngày nay, phần lớn thời gian của chúng ta là dành cho việc sử dụng các thiết bị điện tử. Ví dụ như điện thoại di động, máy tính, tivi … việc chúng ta đang chơi game trên điện thoại hoặc đang sửa bài thuyết trình trên máy tính đó chính là việc chúng ta đang tương tác với thiết bị đó thông qua phần mềm.Vậy phần mềm là gì ?
* Phần mềm là tập hợp dữ liệu hoặc các câu lệnh hướng dẫn, điều khiển cho máy tính để máy tính có thể làm việc theo cách mà bạn muốn. Và người đầu tiên đưa ra lý thuyết về phần mềm vào năm 1935 là Alan Turing (cha đẻ của ngành khoa học máy tính và công nghệ phần mềm).

****

(Hình 1.1: Chân dung của Alan Turing)

Alan Turing sinh ngày 23/6/1912 tại Paddington, London, nước Anh. Từ những năm 1920, Hitler đã tập hợp nhiều chuyên gia để tạo ra loại máy trao đổi thông tin mật có tên là Enigma. Hitler ca ngợi mã Enigma là “Mật mã số một thế giới, cả đến thần thánh cũng không giải nổi”. Năm 1939, Turing dùng trí tuệ của mình để giải mã Enigma góp phần vào cuộc kháng chiến của nhân dân Anh chống phát xít Đức xâm lược. Turing đã sáng chế ra một chiếc máy tính cơ-điện tử giúp giải mã Enigma và được đặt tên là Bombe chiếc máy tính hiện đại đầu tiên của nhân loại. Nhờ vậy mà mọi nội dung mật điện của Đức đều bị quân đội Anh nắm được, tạo ra thế chủ động cho phía Anh. Do vậy mà Thế Chiến thứ 2 đã được kết thúc sớm trước 2 năm và nếu không có ông cục diện chiến tranh lúc đó sẽ khác hẳn. Alan Turing chính là người đặt nền móng cho ngành khoa học máy tính và công nghệ phần mềm nhờ vậy mà chúng phát triển mạnh mẽ sau này.

****

(Hình 1.2: Chiếc máy tính cơ-điện Bombe)

Năm 1975, Tạp chí Popular Electronics vào số tháng một đã giới thiệu máy tính Altair 8080 dưới biệt danh “máy tính mini đầu tiên trên thế giới”. Chiếc máy tính này không có chuột mà điều khiển thông qua các công tắc hoặc nhập liệu bằng bàn phím. Hai người là Paul Allen và Bill Gates đã đề nghị được viết phần mềm cho Altair bằng ngôn ngữ lập trình BASIC. Vào ngày 4 tháng 4, với sự thành Paul Allen và Bill Gates đã thành lập nên công ty phần mềm Microsoft là hãng sản xuất phần mềm lớn nhất thế giới.

****

(Hình 1.3: Paul Allen và Bill Gates - hai nhà sáng lập Microsoft)

**Năm 1977,** Steve Jobs và Steve Wozniak thành lập Apple và giới thiệu máy tính Apple II tại triển lãm máy tính West Coast đầu tiên. Apple II mang tới trải nghiệm đồ hoạ màu sắc, âm thanh, có bộ nhớ RAM và hệ điều hành Apple DOS. Apple một trong năm [công ty lớn](https://vi.wikipedia.org/wiki/B%E1%BB%91n_c%C3%B4ng_ty_c%C3%B4ng_ngh%E1%BB%87_l%E1%BB%9Bn) của ngành [công nghệ thông tin](https://vi.wikipedia.org/wiki/C%C3%B4ng_ngh%E1%BB%87_th%C3%B4ng_tin) [Hoa Kỳ](https://vi.wikipedia.org/wiki/Hoa_K%E1%BB%B3) chuyên [thiết kế](https://vi.wikipedia.org/wiki/Thi%E1%BA%BFt_k%E1%BA%BF), [phát triển](https://vi.wikipedia.org/wiki/Ph%C3%A1t_tri%E1%BB%83n) và [bán](https://vi.wikipedia.org/wiki/B%C3%A1n) [thiết bị điện tử tiêu dùng](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_t%E1%BB%AD_ti%C3%AAu_d%C3%B9ng), [phần mềm máy tính](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m) và các dịch vụ trực tuyến. Có thể kể đến các sản phẩm nổi tiếng như iPod, iPhone, iPad, và các dịch vụ như iTunes, App Store và iCloud, định hình cách chúng ta tương tác với công nghệ trong thế giới hiện đại.



(Hình 1.4: Steve Jobs và Steve Wozniak – 2 nhà sáng lập Apple)

* Năm 1998, Larry Page và Sergey Brin thành lập Google  là một công ty [công nghệ](https://vi.wikipedia.org/wiki/C%C3%B4ng_ngh%E1%BB%87) [đa quốc gia](https://vi.wikipedia.org/wiki/C%C3%B4ng_ty_%C4%91a_qu%E1%BB%91c_gia) của [Mỹ](https://vi.wikipedia.org/wiki/Hoa_K%E1%BB%B3), chuyên về các dịch vụ và sản phẩm liên quan đến [Internet](https://vi.wikipedia.org/wiki/Internet), bao gồm các công nghệ [quảng cáo trực tuyến](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3ng_c%C3%A1o_tr%E1%BB%B1c_tuy%E1%BA%BFn), [điện toán đám mây](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_to%C3%A1n_%C4%91%C3%A1m_m%C3%A2y), [phần mềm](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m), [phần cứng](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_c%E1%BB%A9ng) và nổi tiếng nhất là [công cụ tìm kiếm](https://vi.wikipedia.org/wiki/C%C3%B4ng_c%E1%BB%A5_truy_v%E1%BA%A5n_d%E1%BB%AF_li%E1%BB%87u) phổ biến nhất thế giới Google. Nó giúp chúng ta tìm kiếm thông tin từ hàng ngàn trang web chỉ bằng cách nhập một vài từ khóa. Một số ứng dụng khác như Gmail, Google Maps, Youtube … những phần mềm giúp ích rất nhiều trong cuộc sống của chúng ta ngày nay.



(Hình 1.5:Larry Page và Sergey Brin – 2 nhà sáng lập Google  )

* Năm 2004, **Facebook** là [phương tiện truyền thông xã hội](https://vi.wikipedia.org/wiki/Truy%E1%BB%81n_th%C3%B4ng_x%C3%A3_h%E1%BB%99i) và [dịch vụ mạng xã hội](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_v%E1%BB%A5_m%E1%BA%A1ng_x%C3%A3_h%E1%BB%99i) trực tuyến lớn nhất thế giới thành lập vào năm 2004 của [Mỹ](https://vi.wikipedia.org/wiki/M%E1%BB%B9) thuộc sở hữu của [Meta Platforms](https://vi.wikipedia.org/wiki/Meta_(c%C3%B4ng_ty)) có trụ sở tại [Menlo Park, California](https://vi.wikipedia.org/wiki/Menlo_Park,_California). Nó được [Mark Zuckerberg](https://vi.wikipedia.org/wiki/Mark_Zuckerberg), cùng với các sinh viên [Đại học Harvard](https://vi.wikipedia.org/wiki/%C4%90%E1%BA%A1i_h%E1%BB%8Dc_Harvard) và bạn cùng phòng là [Eduardo Saverin](https://vi.wikipedia.org/wiki/Eduardo_Saverin), [Andrew McCollum](https://vi.wikipedia.org/wiki/Andrew_McCollum), [Dustin Moskovitz](https://vi.wikipedia.org/wiki/Dustin_Moskovitz), [Chris Hughes](https://vi.wikipedia.org/wiki/Chris_Hughes) sáng lập. Facebook ra đời nhằm kết nối tất cả mọi người trên toàn thế giới lại với nhau. Khi dùng ứng dụng này, bạn hoàn toàn có thể gặp gỡ hàng trăm, hàng nghìn bạn bè trên toàn thế giới thông qua mạng Internet. Giống như với mạng Internet thì Facebook sẽ giúp mọi người xóa bỏ mọi khoảng cách địa lý. Bạn có thể đăng và chia sẻ trạng thái, cập nhật hồ sơ cá nhân cũng như tương tác với người khác. Bên cạnh đó, trên mạng xã hội còn có rất nhiều tiện ích khác như bạn sử dụng Facebook để bán hàng và kiếm tiền một cách hiệu quả.



(Hình 1.6: [Mark Zuckerberg](https://vi.wikipedia.org/wiki/Mark_Zuckerberg) – nhà sáng lập Facebook)

* Và đây chính là một trong những người sáng lập và định hình ngành công nghiệp phần mềm hiện đại.



(Hình 1.7: Những người tiên phong của ngành công nghiệp phần mềm hiện đại)

* 1. **Quy trình phát triển phần mềm (Khoảng 6 trang)**
* Cũng như mọi ngành sản xuất khác, quy trình là một trong những yếu tố cực kỳ quan trọng đem lại sự thành công cho các nhà sản xuất phần mềm, nó giúp mọi thành viên trong dự án từ người cũ cho đến người mới, trong hay ngoài công ty đều có thể xử lý đồng bộ công việc tương ứng vị trí của mình qua cách thức chung của công ty, hay ít nhất ở cấp độ dự án.
* Có thể nói quy trình phát triển/xây dựng phần mềm(SDLC - Software Development Life Cycle) có tính chất quyết định để tạo ra sản phẩm chất lượng tốt với chi phí thấp và năng suất cao, điều này có ý nghĩa quan trọng đối với các công ty sản xuất hay gia công phần mềm củng cố và phát triển với nền công nghiệp phần mềm đầy cạnh tranh.



(Hình 1.8: Quy trình phát triển phần mềm - SDLC - Software Development Life Cycle)

**1.2.1 Khái niệm quy trình phát triển phần mềm**

**-**  Quy trình phát triển phần mềm (SDLC - Software Development Life Cycle) là toàn bộ quá trình xây dựng lên sản phẩm đáp ứng các thông số kỹ thuật và yêu cầu của người dùng. SDLC cung cấp một tiêu chuẩn quốc tế mà các công ty sản xuất phần mềm có thể sử dụng để xây dựng và cải tiến sản phẩm công nghệ. Một quy trình tốt sẽ luôn tạo ra những sản phẩm đạt tiêu chuẩn.

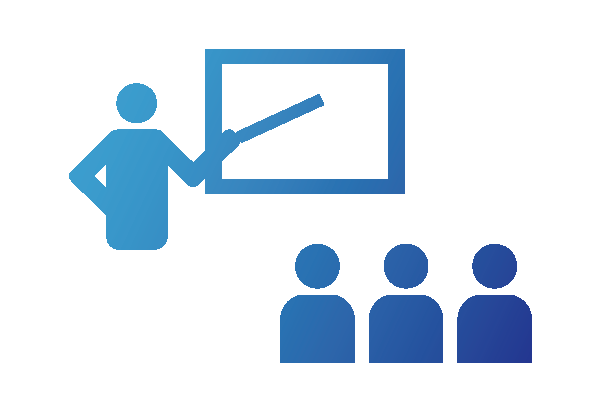
**-** Quy trình được chia thành 6 bước và mỗi giai đoạn đều có sự tham gia của đội ngũ phát triển phần mềm. Quy trình giúp tương tác hóa các hoạt động và yếu tố với nhau một cách nhịp nhàng, đem lại hiệu quả trong quá trình sản xuất phần mềm. Chúng ta hãy cùng tìm kiểu rõ hơn về 6 bước này trong phần tiếp theo.

**1.2.2 Các bước phát triển phần mềm**

**Giai đoạn 1: Needs identification (Xác định nhu cầu)**

### ****Needs identification**** là giai đoạn nghiên cứu thị trường và brainstorming (phương pháp động não) của quy trình. Trước khi xây dựng phần mềm, công ty cần thực hiện nghiên cứu sâu rộng thị trường để xác định khả năng tồn tại của sản phẩm. Developers phải xác định được các chức năng và dịch vụ mà phần mềm nên cung cấp được cho người tiêu dùng để họ cảm thấy sản phẩm cần thiết và hữu ích.

Ngoài ra, developers cũng nên thảo luận cùng với các bộ phận khác trong công ty về : Điểm mạnh, điểm yếu và cơ hội của sản phẩm. Quá trình phát triển phần mềm chỉ bắt đầu nếu sản phẩm thỏa mãn được mọi thông số nhất thiết để thành công.

**

(Hình 1.9: Xác định nhu cầu là giai đoạn 1 trong quy trình phát triển phần mềm)

**Giai đoạn 2: Requirements Analytics (Phân tích yêu cầu)**

**Requirements Analytics** là giai đoạn thực hiện khảo sát chi tiết yêu cầu, mong muốn của khách hàng. Sau đó, thông tin sẽ được tổng hợp vào tài liệu đặc tả yêu cầu ( Prototype). Tài liệu đặc tả phải đầy đủ các yêu cầu về chức năng, phi chức năng và giao diện. Ngoài ra, tài liệu còn cung cấp một bản phác thảo chi tiết về thành phần, phạm vi, nhiệm vụ của developers và các thông số thử nghiệm để tạo ra sản phẩm chất lượng.

Phân tích yêu cầu cũng là giai đoạn mà các developers lựa chọn cách tiếp cận phát triển phần mềm như: Mô hình chữ V (V Model) hay mô hình thác nước (Waterfall)

**Giai đoạn 3: Design (Thiết kế)**

Sau khi đã xác định & phân tích kỹ lưỡng về yêu cầu, chúng ta sẽ chuyển sang giai đoạn nắm vai trò quan trọng thiết yếu của Quy trình phát triển phần mềm - **Design** (thiết kế). Tại đây, các kiến trúc sư và nhà phát triển phần mềm sẽ đưa ra các thông số kỹ thuật tiên tiến mà họ cần để tạo ra sản phẩm theo yêu cầu. Vấn đề cần được thảo luận thêm giữa các bên bao gồm: Mức độ rủi ro, thành phần nhóm, công nghệ áp dụng, thời gian, ngân sách, giới hạn của dự án, phương pháp và thiết kế kiến trúc.

Tài liệu DSD (Đặc điểm kỹ thuật thiết kế) sẽ là kết quả cuối cùng của giai đoạn. DSD chỉ định thiết kế kiến trúc, thành phần, giao tiếp, đại diện front-end và luồng người dùng của sản phẩm.

**Giai đoạn 4: Development (Lập trình)**

Tại giai đoạn 4, [developers](https://itnavi.com.vn/blog/dev-la-nghe-gi) sẽ lập trình và triển khai thông số thiết kế. Lập trình viên sẽ coding dựa trên các thông số kỹ thuật và yêu cầu của sản phẩm đã được thống nhất trong các giai đoạn trước.

Sau khi coding hoàn tất, developers sẽ deploy sản phẩm trong môi trường phát triển (development environment). Lập trình viên sẽ thử nghiệm phiên bản đã tạo ra và điều chỉnh lại cho phù hợp với yêu cầu.

**

Lập trình là giai đoạn thứ 4 của quy trình phát triển phần mềm.

**Giai đoạn 5: Testing (Kiểm thử)**

Sau khi developers đã hoàn thành giai đoạn lập trình, tester sẽ tiếp nhận sản phẩm và tiến hành testing. Tester sẽ tạo test case (Kịch bản kiểm thử) dựa trên tài liệu giải pháp tạo ở giai đoạn 2 và tiến hành kiểm tra. Tester sẽ cập nhật kết quả test vào tool quản lý và thông báo bug (lỗi) đến developers. Tester và developers sẽ cùng nhau phối hợp xử lý các bug và cập nhật trên hệ thống quản lý lỗi. Trong thực tế, tùy theo mô hình phát triển phần mềm mà hoạt động Develop và Kiểm Thử có thể diễn ra song song hoặc tiến hành lần lượt. Vì dụ như ở mô hình Waterfall, lập trình được thực hiện xong mới đến giai đoạn kiểm thử.

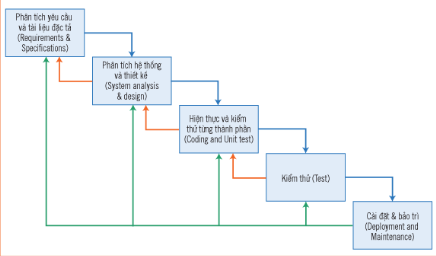
**Giai đoạn 6: Deployment & Maintenance (Triển khai & bảo trì)**

Tại giai đoạn này khi lỗi đã được xử lý xong, nhà phát triển phần mềm sẽ cung cấp sản phẩm hoàn chỉnh đến tay khách hàng. Testing vẫn được diễn ra ở giai đoạn triển khai để đảm bảo sản phẩm luôn có mức độ hoàn hảo cao. Sau khi phát hành, công ty sẽ tạo ra một nhóm bảo trì để quản lý các vấn đề mà khách hàng gặp phải khi sử dụng sản phẩm. Bảo trì giúp khắc phục nhanh các vấn đề nhỏ xảy ra trong quá trình sử dụng sản phẩm.

1.2.3 Các mô hình phát triển phần mềm

a. Mô hình thác nước(Waterfall)

Mô hình thác nước hay còn gọi là mô hình tuyến tính hay mô hình kinh điên (classic model). Trong mô hình này, quy trình phát triển trông giống như một dòng chảy, với các giai đoạn được thực hiện theo trật tự nghiêm ngặt và không có sự quay lui hay nhây vượt giai đoạn. Cụ thể, quá trình sẽ được thực hiện tuần tự qua các giai đoạn: phân tích yêu cầu, thiết kế, triển khai thực hiện, kiểm thứ, liên kết và bảo trì. Tức là mô hình này sẽ xem quá trình xây dựng một sản phầmphần mềm bao gồm nhiều giai đoạn tách biệt, sau khi hoàn tất một giai đoạn thì chuyền đến giai đoạn sau. Sau khi phân tích yêu cầu thì đặc tả yêu cầu, đưa cho khách hàng xem xét. Nếu khách hàng chấp nhận thì sẽ tiến hành tuần tự một loạt các giai đoạn như trên. Chi đưa sản phẩm cho khách hàng khi đã hoàn thiện.



(Mô hình thác nước được coi là mô hình phát triển phần mềm đầu tiên được sử dụng)

Ưu điểm của mô hình Waterfall:

1. Dễ quản lý: Mô hình Waterfall có cấu trúc rõ ràng và dễ quản lý. Mỗi giai đoạn của quy trình phát triển được thực hiện một cách tuần tự, điều này giúp dễ dàng theo dõi tiến độ và quản lý dự án.

2. Tài liệu đầy đủ: Vì mỗi giai đoạn được hoàn thành trước khi chuyển sang giai đoạn tiếp theo, có nhiều tài liệu được tạo ra trong mỗi giai đoạn. Điều này giúp việc hiểu và bảo trì dự án ở các giai đoạn sau.

3. Kiểm soát tốt: Do mỗi giai đoạn được kiểm tra kỹ lưỡng trước khi tiến xa hơn, việc xác định và sửa lối ở giai đoạn trước đó trở nên dễ dàng hơn.

4. Dễ dàng đo lường tiến độ: Bởi vì mối giai đoạn được đặt ra trước với các mục tiêu rõ ràng, việc đo lường tiến độ so với các mục tiêu này trở nên đơn giản.

Nhược điểm của mô hình Waterfall:

1. Khó thích ứng với thay đối: Mô hình Waterfall không linh hoạt. Nếu có yêu cầu mới hoặc thay đổi, việc thực hiện chúng sau khi đã bắt đầu giai đoạn triển khai có thể rất khó khăn.

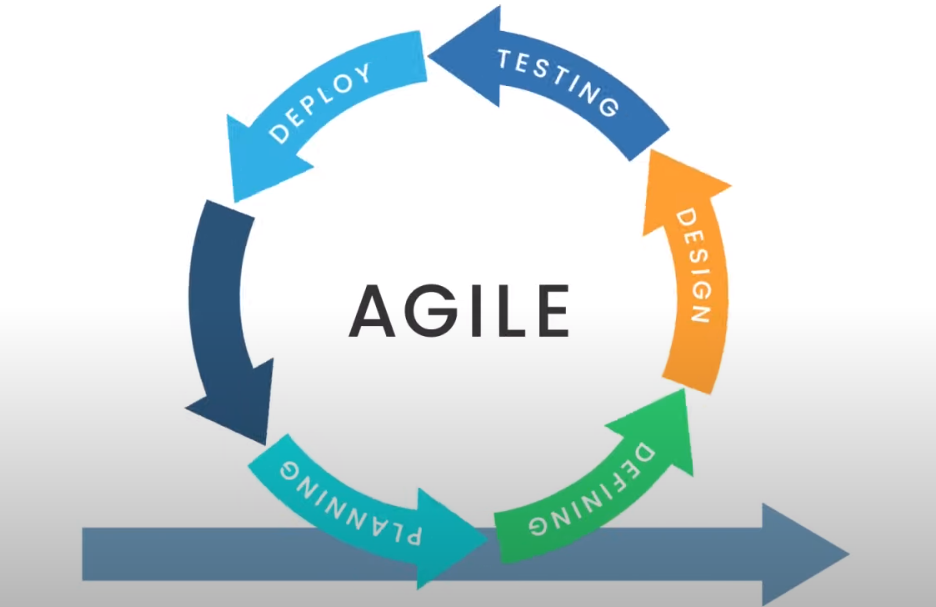
2. Không phản hồi linh hoạt: Do tính tuần tự của Waterfall, không có phản hồi ngay lập tức từ khách hàng hoặc người dùng cho đến khi sản phẩm hoàn thành. Điều này có thể dấn đến việc phát hiện lối hoặc sự không hải lòng của khách hàng ở giai đoạn cuối cùng, khi việc sửa lỗi có thể trở nên đắt đỏ và khó khăn.

3. Khó quản lý dự án lớn: Trong các dự án lớn, việc chia nhỏ dự án thành các giai đoạn có thể trở nên phức tạp và khó duy trì quản lý, đặc biệt nếu có nhiều nhóm làm việc cùng một lúc.

4. Thời gian và chi phí: Nếu có lối phát sinh ở giai đoạn cuối, việc sửa chúng có thể tốn kém hơn nếu lối được phát hiện và sửa ngay từ giai đoạn đầu. Điều này có thể dẫn đến vượt quá ngân sách và thời gian dự định của dự án.

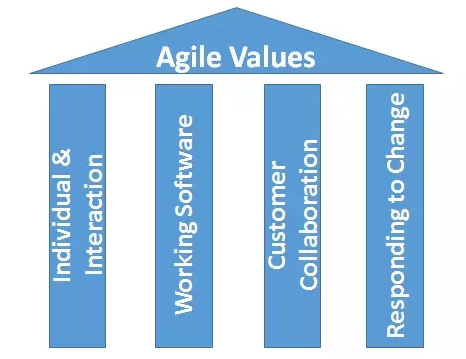
b Mô hình Agile

Khái niệm Agile (Agile Software Development) là một triết lý hoặc một khung tư duy để nhanh chóng thích ứng và phản hồi với sự thay đổi từ đó đạt được thành công trong một môi trường liên tục biến động, không chắc chắn. Phương pháp này là sự phát triển phần mềm linh hoạt thay vì phải dùng phương pháp truyền thống Waterfall. Được ứng dụng trong quy trình phát triển phần mềm với mục tiêu là đưa sản phẩm đến tay người dùng càng nhanh càng tốt, với chất lượng ngày càng được đảm bảo.



Trên thực tế, phương pháp Agile được xây dựng trên nguyên tắc phân đoạn vòng lặp (iterative) và tăng trưởng (incremental). Nó là cách để quản lý một dự án bằng cách chia nó thành nhiều giai đoạn. Nó liên quan đến sự hợp tác liên tục với các bên liên quan và cải tiến liên tục ở mọi giai đoạn. Khi công việc bắt đầu, các nhóm luân chuyển qua một quá trình lập kế hoạch, thực hiện và đánh giá. Sự hợp tác liên tục là rất quan trọng, với cả các thành viên trong nhóm và các bên liên quan của dự án.

* Bốn giá trị cốt lỗi của Agile :



### 1. Cá nhân và sự tương tác qua các quy trình và công cụ

Giá trị đầu tiên trong Tuyên ngôn Agile là “Các cá nhân và sự tương tác qua các quy trình và công cụ”. Đánh giá cao con người hơn các quy trình hoặc công cụ là điều dễ hiểu bởi vì chính những người đáp ứng nhu cầu kinh doanh và thúc đẩy quá trình phát triển. Nếu quy trình hoặc công cụ thúc đẩy sự phát triển, thì nhóm sẽ ít phản ứng hơn với sự thay đổi và ít có khả năng đáp ứng nhu cầu của khách hàng hơn. Giao tiếp là một ví dụ về sự khác biệt giữa đánh giá cá nhân và quá trình. Trong trường hợp cá nhân, giao tiếp là linh hoạt và xảy ra khi có nhu cầu. Trong trường hợp của quá trình, giao tiếp được lên lịch và yêu cầu nội dung cụ thể.(Sẽ không thấy manager, leader) mà các cá nhân sẽ tự tương tác với nhau chủ động trao đổi và làm việc với nhau.

### 2. Phần mềm làm việc trên tài liệu toàn diện

Trong lịch sử, rất nhiều thời gian được dành cho việc ghi chép sản phẩm để phát triển và phân phối cuối cùng. Đặc điểm kỹ thuật, yêu cầu kỹ thuật, bản cáo bạch kỹ thuật, tài liệu thiết kế giao diện, kế hoạch thử nghiệm, kế hoạch tài liệu và phê duyệt cần thiết cho từng loại. Danh sách này rất rộng và là nguyên nhân dẫn đến sự chậm trễ kéo dài trong quá trình phát triển. Agile không loại bỏ tài liệu, nhưng nó sắp xếp hợp lý nó theo một hình thức cung cấp cho nhà phát triển những gì cần thiết để thực hiện công việc mà không bị sa lầy vào những chi tiết vụn vặt. Các yêu cầu về tài liệu Agile dưới dạng câu chuyện của người dùng, đủ để một nhà phát triển phần mềm bắt đầu nhiệm vụ xây dựng một chức năng mới. Tuyên ngôn Agile coi trọng tài liệu, nhưng nó coi trọng phần mềm hoạt động hơn. Các thành viên sẽ dành thời gian nhiều hơn để trao đổi với nhau cách thức vận hành phần mềm hiệu quả hơn thay vì dành quá nhiều thời gian ghi lại những ghi chú không cần thiết. Và đưa ra các tài liệu khô han bắt các thành viên trong team đọc đi đọc lại.

### 3. Hợp tác với khách hàng trong quá trình đàm phán hợp đồng

Đàm phán là khoảng thời gian khách hàng và người quản lý sản phẩm tìm ra các chi tiết của việc giao hàng, với các điểm mà các chi tiết có thể được thương lượng lại. Hợp tác hoàn toàn là một sinh vật khác. Với các mô hình phát triển như Waterfall, khách hàng thương lượng các yêu cầu đối với sản phẩm, thường rất chi tiết, trước khi bắt đầu bất kỳ công việc nào. Điều này có nghĩa là khách hàng đã tham gia vào quá trình phát triển trước khi quá trình phát triển bắt đầu và sau khi hoàn thành, nhưng không phải trong quá trình này. Tuyên ngôn Agile mô tả một khách hàng đã tham gia và cộng tác trong suốt quá trình phát triển. Điều này giúp cho việc phát triển đáp ứng nhu cầu của khách hàng trở nên dễ dàng hơn rất nhiều. Các phương pháp Agile có thể bao gồm khách hàng trong các khoảng thời gian cho các bản trình diễn định kỳ.

### 4. Ứng phó với sự thay đổi theo kế hoạch

Phát triển phần mềm truyền thống coi thay đổi là một khoản chi phí, vì vậy nó phải được tránh. Mục đích là phát triển các kế hoạch chi tiết, phức tạp, với một tập hợp các tính năng được xác định và với mọi thứ, nói chung, có mức độ ưu tiên cao như mọi thứ khác và với một số lượng lớn nhiều phụ thuộc vào việc phân phối theo một thứ tự nhất định để nhóm có thể làm việc trên mảnh tiếp theo của câu đố.

Với Agile, độ ngắn của một lần lặp có nghĩa là các ưu tiên có thể được chuyển từ lần lặp này sang lần lặp khác và các tính năng mới có thể được thêm vào lần lặp tiếp theo. Quan điểm của Agile là những thay đổi luôn cải thiện một dự án; thay đổi cung cấp giá trị bổ sung.

Có lẽ không có gì minh họa cách tiếp cận tích cực của Agile để thay đổi tốt hơn khái niệm Điều chỉnh phương pháp, được định nghĩa trong Phương pháp phát triển hệ thống thông tin Agile đang được sử dụng như: “Một quy trình hoặc khả năng trong

đó các tác nhân của con người xác định cách tiếp cận phát triển hệ thống cho một tình huống dự án cụ thể thông qua các thay đổi đáp ứng và xen kẽ động giữa các bối cảnh, ý định và các đoạn phương pháp. " Các phương pháp Agile cho phép nhóm Agile sửa đổi quy trình và làm cho nó phù hợp với nhóm hơn là ngược lại.

* Ưu điểm :

Phương pháp tiếp cận thích ứng cao

Hạn chế các rủi ro xuống mức thấp

Giao tiếp tức thời, trực tiếp và rõ ràng

Có thể release thường xuyên và dễ dàng thu thập feedback

Có thể release sớm một sản phẩm đang hoạt động

Phát hiện sớm các khiếm khuyết và sửa chữa với chi phí thấp

Đảm bảo tốt sự hài lòng của khách hàng

* Nhược :

Không lý tưởng cho người mới bắt đầu và nhà phát triển thiếu kinh nghiệm

Thiếu hiệu quả về mặt tài liệu

Cần được đào tạo chuyên sâu để hiểu quy trình dự án và áp dụng

Cần chú ý liên tục, nếu không, dự án phát triển nhanh có thể bị lạc mất tiến độ.

**Các pha trong công nghệ phần mềm**

**1.3.1 Quy trình là gì**

**1.3.2 SEP, ISO, CMM/CMMI**

**1.3.3 Các mô hình phân tích từng cái ra**

**1.3 Mô hình hiện đại Degin partern(mvc,rest)**

**Chương 2 : Kiến trúc REST và RESTful API**

Sau khi em đã nói qua Architectural Styles ở chương 1(1.2.4).Em sẽ nói cụ thể về REST

2.1 Sự bắt nguồn của REST

2.2 Các phần tử trong kiến trúc REST

2.3 Góc nhìn kiến trúc REST

Api là gì front end back end

2.4 Kiến trúc RESTful API ( Nguyên lý hoạt động, các ứng dụng …)

**Chương 3 : Ứng dụng của RESTful API cho dữ liệu IoT**

**3.1 Lựa chọn thiết bị ( cảm biến nhiệt độ, độ ẩm …)**

**3.2 Thiết kế cơ sở dữ liệu (MySql)**

**Tầng ứng dụng**

**Moogodb(schema) lưu nhiệt độ trong phòng số mấy nhiệt độ bao nhiêu**

**Viết api để crud**

**3.3 Phát triển Restful Api bằng Spring Boot**

**3.4 Triển khai và giám sát**

**Phát triển để phân tích dữ liệu (nhiệt độ trung bình,viết them hàm)**

**Thầy xem các nội dung chi tiết em định làm trong đồ án ạ. Giúp em chỉ ra những nội dung chưa đúng hoặc thiếu ạ. Chương 3 em chưa có ý tưởng làm ạ, viết api, triển khai ra sao em chưa hiểu rõ ạ. Mong thầy cho em những ý tưởng làm ạ.**

**Đây là tài liệu em tham khảo ạ :**

* Architectural Styles and the Design of Network-based Software Architectures”, được xuất bản vào năm 2000