1. Mô hình Agile:
2. Agile là gì?

* Agile là 1 thuật ngữ có thể hiểu như 1 triết ký hay khung tư tưởng để các doanh nghiệp nhanh chóng thích ứng với các thay đổi trong môi trường kinh doanh biến động, nhiều thay đổi và có phản hồi phù hợp
* Mô hình Agile được xuất hiện trong ngành công nghệ thông tin như 1 phương pháp giúp phát triển phần mềm linh hoạt và nhạy bén, nó tập trung vào sự tương tác giữa các thành viên trong nhóm và thay đổi linh hoạt trong quá trình phát triển. Agile tuân theo 4 tôn chỉ và 12 nguyên tắc như sau:

+ 4 tôn chỉ:

* Individuals and interactions over processes and tools: (Con người và tương tác hơn quy trình và công cụ). Agile coi con người và tương tác giữa các thành viên trong nhóm là yếu tố quan trọng nhất trong phát triển phần mềm
* Working software over comprehensive documentation: (Phần mềm hoạt động hơn là tài liệu chi tiết). Agile tập trung vào việc sản xuất phần mềm hoạt động 1 cách nhanh chóng và linh hoạt hơn là việc tạo ra tài liệu chi tiết về phần mềm
* Customer collaboration over contract negotiation: (Sự hợp tác với khách hàng hơn là đàm phán hợp đồng). Hiểu được khách hàng cần gì để tư vấn và điều chỉnh sản phẩm thay vì chỉ dựa vào các điều khoản trong hợp đồng
* (Phản hồi thay đổi hơn là bám sát kế hoạch): Agile khuyến khích thích nghi với sự thay đổi, đó có thể là thay đổi về công nghệ

+ 12 nguyên tắc quan trọng:

* Đáp ứng toàn diện nhu cầu khách hàng thông qua việc giao hàng sớm và sản phẩm có giá trị
* Thay đổi yêu cầu được chào đón, thậm chí là rất muộn trong quá trình phát triển.
* Giao phần mềm chạy được cho khách hàng một cách thường xuyên
* Nhà kinh doanh và các kỹ sư phần mềm cần làm việc cùng nhau trong suốt dự án
* Xây dựng dự án xung quanh các cá nhân có động lực. Cung cấp sự hỗ trợ cần thiết, môi trường làm việc và niềm tin để hoàn thành công việc
* Trao đổi trực tiếp là cách truyền đạt thông tin hiệu quả nhất
* Thước đo chính của tiến độ là phần mềm chạy tốt.
* Phát triển liên tục và bền vững
* Cải tiến sự linh hoạt bằng cách quan tâm đến kỹ thuật và thiết kế
* Nghệ thuật tối đa hóa lượng công việc chưa xong - Sự đơn giản là cần thiết
* Nhóm tự tổ chức
* Thích ứng thường xuyên với những thay đổi.
* Đặc trưng của Agile:

+ Tính lặp (Iterative): Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại (Iteration hoặc Sprint), thường có khung thời gian ngắn (từ 1-4 tuần). Trong mỗi phân đoạn, nhóm phát triển thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai, kiểm thử để cho ra các phần nhỏ của sản phẩm

+ Tính tăng trưởng và tiến hóa (Incremental & Evolutionary): Cuối các phân đoạn, nhóm cho ra các phần nhỏ của sản phẩm cuối cùng, thường là đầy đủ, có khả năng chạy tốt, được kiểm thử cẩn thận và có thể sử dụng. Theo thời gian, phân đoạn này tiếp nối phân đoạn kia, các phần chạy được này sẽ được tích lũy, lớn dần lên cho tới khi toàn bộ yêu cầu của khách hàng được thỏa mãn.

+ Tính thích nghi (adaptive): Do các phân đoạn chỉ kéo dài trong một khoảng thời gian ngắn và việc lập kế hoạch cũng được điều chỉnh liên tục, nên các thay đổi trong quá trình phát triển (yêu cầu thay đổi, thay đổi công nghệ, thay đổi định hướng về mục tiêu v.v.) đều có thể được đáp ứng theo cách thích hợp.

+ Nhóm tự tổ chức và liên chức năng: Các cấu trúc nhóm này tự phân công công việc mà không dựa trên các mô tả cứng về chức danh hay làm việc dựa trên một sự phân cấp rõ ràng trong tổ chức. Nhóm tự tổ chức đã đủ các kĩ năng cần thiết để có thể được trao quyền tự ra quyết định, tự quản lí và tổ chức lấy công việc của chính mình để đạt được hiệu quả cao nhất.

+ Quản lý tiến trình thực nghiệm: (Empirical Process Control): Các nhóm Agile ra các quyết định dựa trên các dữ liệu thực tiễn thay vì tính toán lý thuyết hay các tiền giả định. Agile rút ngắn vòng đời phản hồi để dễ dàng thích nghi và gia tăng tính linh hoạt nhờ đó có thể kiểm soát được tiến trình, và nâng cao năng suất lao động.

+ Giao tiếp trực diện (face-to-face communication): Agile không phản đối việc tài liệu hóa, nhưng đánh giá cao hơn việc giao tiếp trực diện thay vì thông qua giấy tờ. Agile khuyến khích nhóm phát triển trực tiếp nói chuyện để hiểu rõ hơn về cái khách hàng thực sự cần. Trong giao tiếp giữa nội bộ nhóm, Agile khuyến khích trực tiếp trao đổi và thống nhất với nhau về thiết kế của hệ thống và cùng nhau triển khai thành các chức năng theo yêu cầu

+ Phát triển dựa trên giá trị (value-based development): Một trong các nguyên tắc cơ bản của agile là “sản phẩm chạy tốt chính là thước đo của tiến độ”. Nhóm Agile thường cộng tác trực tiếp và thường xuyên với khách hàng để biết yêu cầu nào có độ ưu tiên cao hơn, mang lại giá trị hơn sớm nhất có thể cho dự án.

* Ưu điểm của mô hình Agile:

+ Agile ban đầu được tạo nên cho ngành công nghiệp phát triển phần mềm để giúp cho việc sắp xếp và cải tiến quá trình sản xuất. Qua đó, các nhà phát triển có thể nhận dạng, điều chỉnh các vấn đề và khiếm khuyết một cách nhanh chóng

+ Là một phương pháp thay thế cho cách tiếp cận Waterfall truyền thống, Agile cung cấp phương pháp quản lý giúp các nhóm làm việc cho ra đời một sản phẩm tốt hơn, nhanh hơn thông qua các phiên ngắn và các phiên tương tác /các sprint. Với những kỳ vọng ngày càng gia tăng của khách hàng, việc cạnh tranh liên tục đòi hỏi phải tìm kiếm được những nhà lãnh đạo dự án có thể sử dụng phương pháp tiếp cận tốt nhất để thực hiện dự án.

* Thực hiện thay đổi dễ dàng: Bởi vì dự án được chia thành các thành phần nhỏ, riêng biệt, không phụ thuộc lẫn nhau, nên những thay đổi được thực hiện rất dễ dàng, ở bất kỳ giai đoạn nào của dự án
* Không cần phải nắm mọi thông tin ngay từ đầu: Phù hợp với những dự án chưa xác định được mục tiêu cuối cùng, vì việc này ko quá cần thiết trong giai đoạn đầu
* Bàn giao nhanh hơn: Việc chia nhỏ dự án cho phép đội ngũ có thể tiến hành kiểm tra theo từng phần, xác định và sửa chữa vấn đề nhanh hơn, nhờ đó vc bàn giao công việc sẽ nhất quán và thành công hơn
* Chú ý đến phản hồi của khách hàng và người dùng: Cả khách hàng và người dùng cuối đều có cơ hội để đóng góp ý kiến và phản hồi, từ đó họ sẽ có ảnh hưởng 1 cách mạnh mẽ và tích cực tới sản phẩm cuối cùng
* Cải tiến liên tục: Agile khuyến khích thành viên trong đội ngũ làm việc và khách cung cấp phản hồi của mình, khi đó các giai đoạn khác nhau của sản phẩm cuối có thể được kiểm tra và cải thiện lại nhiều nếu cần
* Nhược điểm của Agile:
* Khó lên kế hoạch dự án: Khá là khó để xác định rõ ràng thời gian bàn giao sản phẩm cuối cùng, vì dự án được chia nhỏ thành các phần khác nhau và mỗi phần lại có thời gian bàn giao riêng biệt.
* Bắt buộc phải hướng dẫn và đào tạo chi tiết:  Phương pháp Agile phức tạp hơn nhiều so với phương pháp truyền thống. Họ sẽ cần phải trải qua đào tạo, hướng dẫn thì mới có thể nắm được phương pháp một cách rõ ràng, đặc biệt là thời gian đầu.
* Ít tài liệu hướng dẫn: Vì Agile thay đổi rất nhiều nên các tài liệu thích hợp cũng thường bị bỏ qua, vì không xác định rõ được kỳ vọng và thành phẩm ngay từ đầu. Mặc dù tài liệu không phải là yếu tố quan trọng nhất, nhưng chúng vẫn rất cần thiết.
* Bắt buộc phải hợp tác để dự án thành công: Điều này đòi hỏi một sự cam kết về thời gian từ cả hai bên trong suốt thời gian của dự án mà các cấu trúc quản lý dự án khác không luôn yêu cầu. Phải có sự tham gia tích cực của người dùng và tiếp tục cộng tác để nó hoạt động.
* Chi phí cao: Chi phí thực hiện theo phương pháp Agile thường hơn một chút so với các phương pháp phát triển khác.
* Các phương pháp Agile: Mô hình Agile Scrum có nhiều phương pháp khác nhau để áp dụng trong 1 dự án. Mỗi phương pháp có những vai trò, chức năng khác nhau tùy vào mục đích và nhu cầu

+ Scrum: Là 1 khung quy trình bao gồm vai trò,sự kiện và quy tắc chung được thực hiện lặp đi lặp lại trong 1 thời gian ngắn (thường là dưới 1 tháng). Một Scrum bao gồm nhiều phòng ban và tổ chức để đẩy mạnh phát triển dự án, sản phẩm

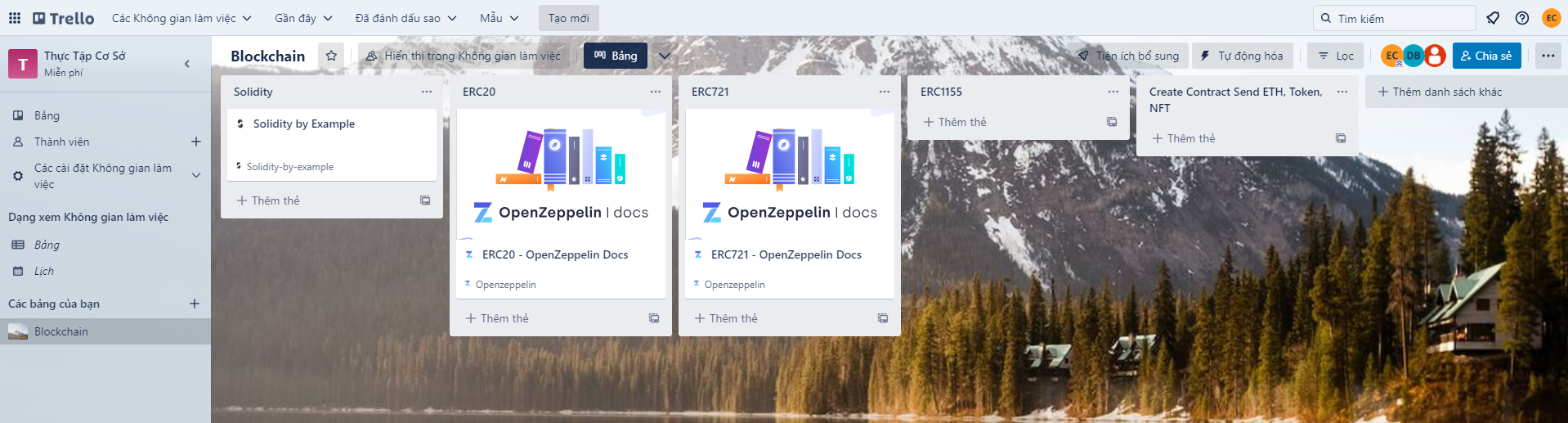
+ Extreme Programming: Đây là phương pháp phát triển dựa trên chu kì nhất định, với sự đơn giảm hóa và cải tiến liên tục trong quá trình thực hiện dự án. Dưới đây là những nguyên tắc cốt lõi của EP: Truyền thông, đơn giản hóa, phản hồi, can đảm, tôn trọng

+ Kanban: Thường được sử dụng trong sản xuất gọn (lean) để kiểm soát tồn kho. Phương pháp này thực hiện dựa trên nguyên tắc chỉ bổ sung khi thực sự cần thiết và quản lý 1 cách trực quan

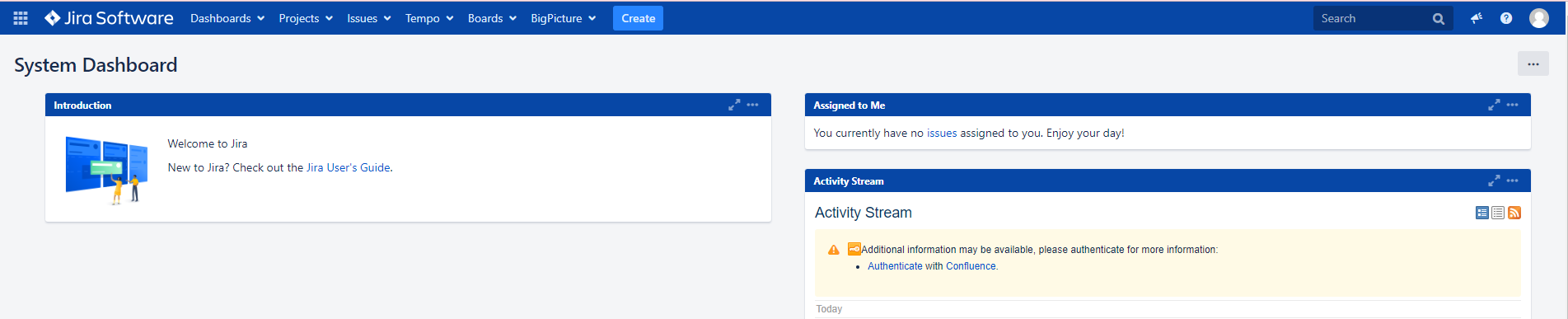
+ Crystal Methodology: Phương pháp này nhận định các công việc như chính sách, thực hành và quy định phải được tùy chỉnh để đáp ứng tính duy nhất của dự án. Giá trị của nó nằm ở con người, sự tương tác, tính cộng đồng, các kĩ năng cần thiết và tính truyền thông

* Một số công cụ được sử dụng để quản lý dự án theo Agile:

+ Trello: Là ứng dụng cho phép người quản lý dự án thông minh theo dạng bảng chứa danh sách. Ứng dụng giúp tạo lập kế hoạch, tổ chức quy trình làm việc và theo dõi tiến độ một cách trực quan, hiệu quả.



+ Jira: Jira cung cấp dịch vụ quản lý theo dõi lỗi, phát hiện các vấn đề của dự án, quản lý hiệu quả quá trình phát triển phần mềm và di động. Những tính năng hữu ích mà phần mềm cung cấp: Quản lý theo dõi dự án, Quản lý các task hoặc các vấn đề, thiết lập quy trình làm việc tương ứng với yêu cầu của từng dự án, cung cấp đa dạng các loại báo cáo, thống kê với nhiều loại biểu đồ khác nhau.



* Quy trình áp dụng Agile vào quản lý dự án:

+ Xác định tầm nhìn dự án

+ Xây dựng lộ trình dự án

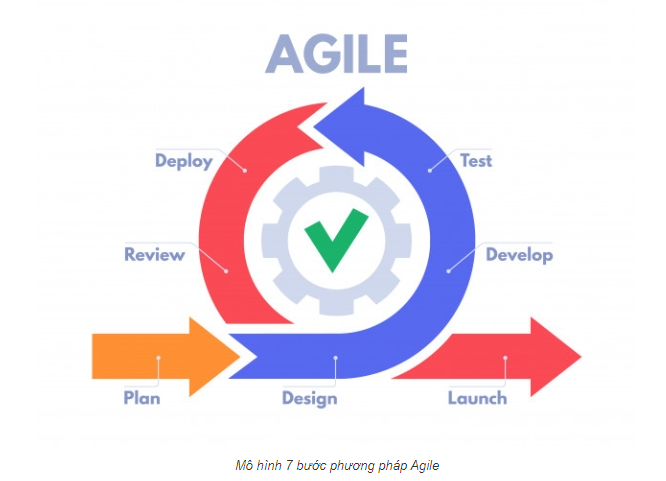
+ Lập kế hoạch giai đoạn

+ Phân tích yêu cầu

+ Triển khai

+ Kiểm thử

+ Bàn giao



1. Mô hình Microservice:

* Mô hình Microservice là 1 kiến trúc phần mềm phân tán mà trong đó ứng dụng được phát triển và triển khai như 1 tập hợp các dịch vụ nhỏ, độc lập và có thể mở rộng. Mỗi dịch vụ đảm nhận 1 chức năng cụ thể trong ứng dụng và có thể được triển khai độc lập với các dịch vụ được kết nối với nhau thông qua các giao thức khác nhau như http, SOA, socket, Massage Queue
* Trước khi mô hình microservice ra đời các ứng dụng thường được phát triển theo mô hình Monolithic architecture (Kiến trúc một khối). Đây là mô hình đưa tất cả các module của project gộp vào làm một. Mô hình kiến trúc một khối thường được chia làm nhiều module nhưng khi đóng gói sẽ cài đặt thành 1 khối. Lợi ích của mô hình này là dễ dàng phát triển và triển khai nhưng nhược điểm của nó là rất khó để bảo trì, tính linh hoạt mở rộng kém. Đó là lí do mà microservice ra đời
* Những đặc điểm của Microservice:

+ Decoupling: Các service trong một hệ thống phần lớn được tách rời. Vì vậy, toàn bộ ứng dịnh có thể dễ dàng được xây dựng, thay đổi và thu nhỏ

+ Componentization: Microservices được coi là các thành phần độc lập có thể dễ dàng thay thế và nâng cấp

+ Business Capabilities: Mỗi thành phần trong kiến trúc microservice rất đơn giản và tập trung vào 1 nhiệm vụ duy nhất

+ Autonomy: Các lập trình viên hay nhóm có thể làm việc độc lập với nhau trong quá trình phát triển

+ Continous Delivery: Cho phép phát hành phần mềm thường xuyên, liên tục

+ Responsibility

+ Decentralized Governance: Không có mẫu chuẩn hóa hoặc bất kỳ mẫu công nghệ nào. Nó được tự do lựa chọn công cụ hữu ích tốt nhất để có thể giai quyết vấn đề

+ Agility: Hỗ trợ phát triển mô hình Agile

* Các nguyên tắc khi phát triển
* Ưu điểm của mô hình:

+ Kiến trúc Microservices được sinh ra để khắc phục những hạn chế của kiến trúc một khối.

* Independent Development - Tất cả các service có thể được phát triển dễ dàng dựa trên chức năng cá nhân của từng service. Có thể chia nhỏ để phát triển độc lập.
* Independent Deployment - Có thể được triển khai riêng lẻ trong bất kỳ ứng dụng nào.
* Fault Isolation - khi một service của ứng dụng không hoạt động, hệ thống vẫn tiếp tục hoạt động.
* Mixed Technology Stack - Các ngôn ngữ và công nghệ khác nhau có thể được sử dụng để xây dựng các service khác nhau của cùng một ứng dụng.
* Granular Scaling

+ Kiến trúc Microservices giúp đơn giản hóa hệ thống, chia nhỏ hệ thống ra làm nhiều service nhỏ lẽ dể dàng quản lý và triển khai từng phần so với kiến trúc nguyên khối. Phân tách rõ ràng giữa các service nhỏ. Cho phép việc mỗi service được phát triển độc lập. Cũng như cho phép lập trình viên có thể tự do chọn lựa technology stack cho mỗi service mình phát triển. mỗi service có thể được triển khai một cách độc lập (VD: Mỗi service có thể được đóng gói vào một docker container độc lập, giúp giảm tối đa thời gian deploy). Nó cũng cho phép mỗi service có thể được scale một cách độc lập với nhau. Việc scale có thể được thực hiện dễ dàng bằng cách tăng số instance cho mỗi service rồi phân tải bằng load balancer.