**1. Software developer**

- Software developer chịu trách nhiệm tạo ra sản phẩm bằng cách viết code bằng [nhiều ngôn ngữ lập trình khác nhau.](https://blog.kms-solutions.asia/most-in-demand-programming-languages-for-finance-and-fintech) Những sản phẩm họ tạo ra có thể là ứng dụng di động, phần mềm máy tính, hoặc bất kỳ sản phẩm phần mềm nào khác để giải quyết một vấn đề nào đó của người dùng.

- Các nhà phát triển thường được phân loại thành:

* Frontend Developer: tạo ra phần giao diện của ứng dụng mà người dùng cuối nhìn thấy và tương tác. Họ tập trung vào khả năng sử dụng và chức năng của sản phẩm phần mềm để đảm bảo người dùng có trải nghiệm tốt.
* Backend Developer là người chịu trách nhiệm phát triển phần "đằng sau" của ứng dụng, nơi dữ liệu được xử lý, lưu trữ và quản lý. Họ làm việc với các ngôn ngữ lập trình như Java, Ruby, hoặc PHP để xây dựng các dịch vụ và ứng dụng server-side. Backend developer tập trung vào xử lý dữ liệu, quản lý cơ sở dữ liệu, đảm bảo tính bảo mật và hiệu suất của hệ thống.
* Fullstack developer: Fullstack developer có khả năng làm việc trên cả phần frontend và phần backend của ứng dụng. Họ có thể xử lý các vấn đề liên quan đến cơ sở dữ liệu và API, xây dựng trang web hướng tới người dùng, làm việc với khách hàng trong giai đoạn lập kế hoạch của dự án,…

**2. Tester**

- Vai trò của tester là kiểm tra, tìm kiếm các lỗi của phần mềm, thẩm định liệu phần mềm đó đã đáp ứng được các yêu cầu kỹ thuật và yêu cầu nghiệp vụ hay không.

- Một số nhiệm vụ của tester:

* Tester phải tham gia vào việc lập kế hoạch kiểm thử từ đầu dự án. Điều này bao gồm việc xác định phạm vi kiểm thử, thiết kế các test case, và lên kế hoạch cho các giai đoạn kiểm thử trong quy trình phát triển.
* Tester phải ghi lại mọi lỗi và sự cố mà họ phát hiện trong quá trình kiểm thử, và sau đó báo cáo chúng cho nhóm phát triển để sửa lỗi. Họ cũng cần theo dõi tiến trình sửa lỗi và kiểm tra lại sau khi lỗi được sửa.
* Tester phải thực hiện kiểm thử hiệu suất để đảm bảo rằng phần mềm hoạt động đúng cách trong các tình huống tải cao và kiểm tra bảo mật để đảm bảo rằng phần mềm không bị xâm nhập hoặc tấn công.

**3. Business Analyst**

- Vai trò của Business Analyst (BA) trong một dự án phát triển phần mềm là tập trung vào việc hiểu và biểu đồ hoá nhu cầu của khách hàng để đảm bảo rằng phần mềm được phát triển đáp ứng đúng các mục tiêu kinh doanh và nhu cầu của dự án

- Một số nhiệm vụ của BA:

* Business Analyst (BA) là người xung phong đầu tiên trong một dự án với vai trò là cầu nối giữa khách hàng và nhóm phát triển. BA sẽ tương tác với khách hàng, người dùng cuối và các bên liên quan khác để thu thập và hiểu yêu cầu của họ, lắng nghe mô tả dự án và biến nó thành yêu cầu cụ thể (như sản phẩm cần thêm, sửa, xóa tính năng gì) và đề xuất các giải pháp kỹ thuật để các nhà phát triển phần mềm có thể sử dụng để xây dựng phần mềm.
* BA đóng một vai trò thiết yếu khi bắt đầu các dự án CNTT vì họ cần thu thập thông tin, phân tích dữ liệu và đảm bảo rằng mọi thành viên trong nhóm đều có hiểu biết chung về những gì đang được xây dựng.
* BA giúp quản lý các yêu cầu thay đổi trong suốt quá trình phát triển và đảm bảo rằng các thay đổi này được quản lý một cách hiệu quả và không ảnh hưởng đến phạm vi và lịch trình của dự án.

**4. UI/UX designer**

**-** Vai trò của UI/UX designer (Thiết kế Giao diện Người dùng/Trải nghiệm Người dùng) trong một dự án phát triển phần mềm là tập trung vào việc tạo ra giao diện người dùng hấp dẫn và trải nghiệm người dùng dễ sử dụng

Một số nhiệm vụ của UI/UX designer:

* Thiết kế giao diện người dùng cho sản phẩm dựa trên danh sách chức năng trên nhiều nền tảng và thiết bị khác nhau, ví dụ như điện thoại di động, máy tính bảng, máy tính,... Giao diện phải được thiết kế sao cho thân thiện, đẹp mắt, dễ tiếp cận và dễ dàng sử dụng với khách hàng, ngay cả khi họ là người mới.
* UI/UX designer phải nắm vững nhu cầu và mong muốn của người dùng cuối bằng các cuộc nghiên cứu và phân tích thị trường. Họ thường thực hiện kiểm tra sử dụng và thu thập phản hồi từ người dùng để đảm bảo rằng giao diện và trải nghiệm người dùng được cải thiện liên tục.
* Trừ khi được hợp nhất thành một vai trò trong các dự án nhỏ hơn, các nhà thiết kế UX và UI chủ yếu tương tác với nhau. Họ cũng làm việc chặt chẽ với các lập trình viên để đảm bảo rằng thiết kế của họ được triển khai một cách đúng đắn và thích hợp.

**5. Project manager**

- Vai trò của một Project Manager (Quản lý dự án) trong một dự án phát triển phần mềm là quản lý và điều hành toàn bộ quá trình dự án để đảm bảo rằng nó được thực hiện một cách hiệu quả và đạt được các mục tiêu kế hoạch và kinh doanh.

- Một số nhiệm vụ của Project manager:

* Project Manager có vai trò xác định những công việc cụ thể cần thực hiện để hoàn thành một dự án và kiểm soát quá trình thực hiện các công việc đó. Họ phải quản lý toàn bộ quá trình dự án, dự đoán và quản lý rủi ro cũng như đưa ra các biện pháp sửa đổi nếu cần thiết để đảm bảo rằng nó được thực hiện một cách hiệu quả.
* Project Manager phải xác định phạm vi dự án, thiết lập mục tiêu và lên kế hoạch chi tiết về lịch trình, nguồn lực và ngân sách.
* Họ là người trung gian trong việc truyền thông giữa các bên liên quan trong dự án, bao gồm khách hàng, nhóm phát triển, người dùng cuối và các bên ngoài khác. Project Manager đảm bảo rằng thông tin liên quan đến dự án được truyền đạt một cách hiệu quả.
* Project Manager phải đánh giá tổng thể về dự án và báo cáo kết quả và tiến trình dự án đối với các bên liên quan, bao gồm cả việc so sánh kết quả với kế hoạch ban đầu.

**6. Security Engineer**

- Vai trò chính của Security Engineer là bảo vệ hệ thống máy tính, mạng lưới, và dữ liệu của tổ chức khỏi các cuộc tấn công mạng và rủi ro bảo mật.

- Một số nhiệm vụ của Security Engineer:

* Security Engineer phải xác định các rủi ro bảo mật tiềm năng trong dự án phát triển. Điều này bao gồm việc xác định các điểm yếu tiềm năng, thiết kế không an toàn và tìm hiểu về các mối đe dọa bảo mật hiện tại.
* Security Engineer tham gia vào việc thiết kế các biện pháp bảo mật, bao gồm việc thiết kế các lớp bảo mật, kiến trúc an toàn và các biện pháp kiểm tra bảo mật để đảm bảo rằng phần mềm được xây dựng với sự bảo mật tích hợp.
* Security Engineer thường thực hiện kiểm tra bảo mật định kỳ để xác định và giải quyết các lỗ hổng bảo mật mật trong dự án phát triển, theo dõi và phát hiện các hoạt động không bình thường trên mạng, và thực hiện các biện pháp phòng ngừa để đảm bảo tính an toàn của hệ thống.

**7. DevOps Engineer**

- Vai trò của một DevOps Engineer trong một dự án phát triển phần mềm liên quan đến việc triển khai và duy trì quy trình tự động hóa và cơ sở hạ tầng để đảm bảo rằng phần mềm được phát triển và vận hành một cách hiệu quả

- Các công việc chính của DevOps Engineer:

* DevOps Engineer giúp tối ưu hóa và tự động hóa quy trình phát triển, từ việc phát triển mã nguồn đến kiểm thử và triển khai. Điều này giúp giảm thiểu thời gian và công sức cần thiết để triển khai các thay đổi và tích hợp chúng vào sản phẩm.
* DevOps Engineer giúp tự động hóa quản lý phiên bản phần mềm, từ việc quản lý mã nguồn đến việc quản lý thay đổi và phiên bản ứng dụng. Điều này giúp đảm bảo sự nhất quán và kiểm soát trong quản lý phiên bản.
* DevOps Developer quản lý cơ sở hạ tầng hệ thống, bao gồm việc cấu hình máy chủ, mạng và lưu trữ. Họ sử dụng các công cụ như Docker, Kubernetes, và các dịch vụ đám mây để quản lý cơ sở hạ tầng hiệu quả và mở rộng dự án khi cần thiết.
* DevOps Developer thực hiện giám sát hệ thống và ứng dụng để theo dõi hiệu suất và phát hiện sự cố sớm. Họ cũng thực hiện các biện pháp để cải thiện hiệu suất và độ ổn định của hệ thống.

**8. Software Architecture**

- Software Architecture đóng một vai trò chủ chốt trong quá trình thiết kế và phát triển phần mềm. Software Architect là người định hình cấu trúc tổng thể của phần mềm và đảm bảo rằng nó đáp ứng được các yêu cầu chức năng, có tính mở rộng cao và ít để lại nợ công nghệ.

- Một số nhiệm vụ của Software Architecture:

* Software Architect phải xác định kiến trúc tổng thể của phần mềm, bao gồm các thành phần chính, tương tác giữa chúng, cách chúng hoạt động cùng nhau và đảm bảo phần mềm thỏa mản các yêu cầu về hiệu suất, bảo mật, mở rộng, và tích hợp hệ thống.
* Software Architecture là người đưa ra thiết kế hệ thống và theo dõi sát nhóm Developer khi xây dựng và vận hành hệ thống, cũng như khi duy trì và mở rộng hệ thống để đảm bảo họ làm theo đúng thiết kế.
* Software Architecture là người lựa chọn công nghệ, ngôn ngữ lập trình, framework và các công cụ phù hợp để triển khai kiến trúc của họ. Lựa chọn này có thể ảnh hưởng đến hiệu suất, tích hợp và khả năng mở rộng của phần mềm.
* Software Architecture phải đảm bảo tính nhất quán trong kiến trúc phần mềm. Điều này đồng nghĩa với việc đảm bảo rằng các phần của hệ thống hoạt động cùng nhau một cách đúng đắn và không xảy ra xung đột.

**9. Data Engineer**

- Vai trò của Data Engineer trong một dự án phát triển phần mềm là thu thập, xử lý, lưu trữ và triển khai dữ liệu trong dự án phần mềm và đảm bảo rằng hệ thống có khả năng xử lý và quản lý dữ liệu một cách hiệu quả, đáng tin cậy và bảo mật.

- Một số nhiệm vụ của Data Engineer:

* Data Engineer phải xác định các nguồn dữ liệu cần thiết cho dự án và thiết lập các quy trình để thu thập dữ liệu từ các nguồn này. Điều này có thể bao gồm dữ liệu từ cơ sở dữ liệu, dịch vụ web, tệp dữ liệu, hoặc các nguồn dữ liệu khác.
* Data Engineer phải thực hiện các biện pháp để làm sạch, biến đổi và chuyển đổi dữ liệu để đảm bảo rằng nó phù hợp với cấu trúc và định dạng cần thiết cho phần mềm. Điều này có thể bao gồm loại bỏ dữ liệu không hợp lệ, kết hợp dữ liệu từ nhiều nguồn, và áp dụng các quy tắc xử lý dữ liệu.
* Data Engineer phải quyết định về việc lưu trữ dữ liệu và triển khai các hệ thống lưu trữ dữ liệu phù hợp. Điều này có thể bao gồm cơ sở dữ liệu SQL, NoSQL, hệ thống tệp hoặc dịch vụ lưu trữ đám mây.
* Data Engineer phải tối ưu hóa hiệu suất của hệ thống xử lý dữ liệu để đảm bảo rằng nó có khả năng xử lý tải cao mà không gặp sự cố.

**Câu 1: Quy Trình Là Gì? Tại sao cần quy trình để xử lý công việc?**

**Quy trình được hiểu cơ bản là cách thức cụ thể để tiến hành một hoạt động hay một quá trình. Quy trình xác định đầu vào, đầu ra của quá trình và cách thức để biến đầu vào thành đầu ra bao gồm việc gì cần phải làm, ai làm, làm lúc nào, ở đâu và như thế nào.**

Nói một cách khác, quy trình là phương pháp được thiết lập để hoàn thành một nhiệm vụ, thường là với các bước được thực hiện theo một thứ tự nhất định.

**Một quy trình có thể nhằm kiểm soát nhiều quá trình, và ngược lại, một quá trình có thể được kiểm soát bằng nhiều quy trình.**

**VD:** Quy trình Xử lý Đơn Hàng Sản Phẩm

Bước 1: Nhận Đơn Hàng

Khách hàng gửi đơn hàng qua trang web hoặc điện thoại.

Nhân viên bán hàng ghi lại thông tin đơn hàng, bao gồm sản phẩm, số lượng, địa chỉ giao hàng và thông tin liên hệ của khách hàng.

Bước 2: Xác nhận Đơn Hàng

Nhân viên bán hàng kiểm tra tính khả thi của đơn hàng, bao gồm kiểm tra số lượng sản phẩm có sẵn trong kho và khả năng giao hàng đúng thời hạn.

Nếu đơn hàng khả thi, đơn hàng được xác nhận và chuyển đến bước tiếp theo. Nếu không, khách hàng được thông báo về tình trạng đơn hàng.

Bước 3: Chuẩn bị Sản Phẩm

Nhân viên kho lấy sản phẩm từ kho và kiểm tra chất lượng.

Sản phẩm được đóng gói cẩn thận, đánh dấu và chuẩn bị cho quá trình vận chuyển.

Bước 4: Vận Chuyển Đơn Hàng

Đơn hàng được chuyển đến bộ phận vận chuyển hoặc công ty vận chuyển bên ngoài.

Thông tin vận đơn và thông tin giao hàng được ghi lại và chuyển đến bộ phận giao hàng.

Bước 5: Giao Hàng

Bộ phận giao hàng tạo lịch giao hàng dựa trên thông tin đơn hàng và địa chỉ giao hàng.

Giao hàng đến địa chỉ của khách hàng, đảm bảo rằng sản phẩm được giao đúng thời hạn và trong tình trạng tốt nhất.

Bước 6: Đánh Giá và Phản Hồi

Khách hàng nhận sản phẩm và được yêu cầu đánh giá chất lượng sản phẩm và dịch vụ.

Phản hồi từ khách hàng được ghi lại và sử dụng để cải thiện quy trình.

Bước 7: Kết Thúc Đơn Hàng

Đơn hàng được đánh dấu là đã hoàn thành sau khi khách hàng đã nhận sản phẩm và đánh giá.

Thanh toán cuối cùng được thực hiện nếu cần thiết.

**Tại sao cần quy trình để xử lý công việc?**

**Giúp công việc được thực hiện nhất quán và đồng bộ:** Trong một tổ chức, chuyện các cá nhân có kiến thức và kỹ năng khác nhau sẽ dẫn đến cách làm việc và hiệu quả khác nhau. Vì vậy nên mới cần có quy trình nhằm mục đích có thể giúp cho các cá nhân đó có thể thực hiện các công việc một cách nhất quán và đồng bộ . Tránh tình trạng nhân viên chậm chỉ thị của cấp trên hay thực hiện các công việc thừa gây lãng phí thời gian.

Đối với các công việc cần phối hợp nhóm thì ngoài sẽ giúp phân bổ công việc hợp lý còn giúp tăng sự phối hợp giữa các thành viên, **đảm bảo công việc được thực hiện đúng tiến độ và trình tự như dự kiến.**

**Quy trình còn giúp cấp trên quản lý tốt nhân viên của mình hơn** thông qua kiểm tra tiến độ làm việc của họ được thể hiện trên quy trình, từ đó đưa ra các giải pháp hay chiến lược kịp thời.

**Giảm rủi ro: Quy trình có thể giúp phát hiện và đối phó với các rủi ro tiềm ẩn. Bằng cách kiểm soát và theo dõi trong quy trình, bạn có thể ứng phó với sự cố hoặc thay đổi nhanh chóng.**

**Tạo khả năng mở rộng: Khi có quy trình, việc mở rộng hoạt động hoặc mở rộng doanh nghiệp trở nên dễ dàng hơn. Bạn có thể dựa vào quy trình hiện có để tạo ra bản sao và triển khai ở các vị trí mới.**

**Câu 2: Quy trình phần mềm là gì? Tại sao cần có quy trình phần mềm trong việc sản xuất phần mềm?**

**Quy trình phần mềm (Software Development Process) là một tập hợp các bước và hoạt động được sắp xếp logic để phát triển và quản lý phần mềm** từ khâu lập kế hoạch, thiết kế, phát triển, kiểm thử cho đến triển khai và bảo trì. **Quy trình phần mềm giúp đảm bảo rằng quá trình phát triển phần mềm diễn ra một cách có hệ thống, hiệu quả và đáp ứng được mục tiêu cụ thể của dự án.**

**Các giai đoạn phát triển phần mềm**

Theo quy tắc chung, quy trình phát triển phần mềm sẽ bao gồm 6 bước:

**Bước 1: Analysis (Lập kế hoạch và phân tích yêu cầu)**

Trước khi bắt đầu xây dựng phần mềm, chúng ta cần thu thập và xác định rõ các yêu cầu của người dùng và các bên liên quan đối với sản phẩm phần mềm sắp xây dựng. Chúng ta cần nghiên cứu thị trường để xác định các chức năng mà phần mềm nên cung cấp cho người dùng để họ cảm thấy đây là phần mềm hữu ích cho họ. Việc nghiên cứu này cũng giúp ta xác định được khả năng tồn tại của phần mềm trên thị trường như thế nào?

Sau đó, các thành viên trong nhóm phát triển phần mềm sẽ làm việc cùng với khách hàng để đưa ra các thông số kỹ thuật và yêu cầu chi tiết về sản phẩm phần mềm dự định làm ra. Tất cả thông tin này sẽ được tổng hợp thành một tài liệu được gọi là tài liệu đặc tả yêu cầu phần mềm (Software Requirement Specification). Tài liệu sẽ bao gồm các yêu cầu về chức năng, giao diện, hiệu suất,… Ngoài ra, còn có cả bản phác thảo về thành phần, nhiệm vụ của từng developer và các thông số thử nghiệm để tạo nên một sản phẩm chất lượng.

Ở giai đoạn này, người quản lý và các nhà phát triển phần mềm sẽ thống nhất việc lựa chọn kiểu mô hình phát triển phần mềm nào

**Bước 2: Design (Thiết kế phần mềm)**

Từ các yêu cầu và thông số kỹ thuật được đưa ra trong bước 1, các nhà phát triển phần mềm sẽ vạch ra kiến trúc tổng thể cần thiết để tạo ra phần mềm. Ngoài ra, các yếu tố như: ngân sách, thời gian, công nghệ áp dụng, mức độ rủi ro,… cũng được xác định rõ ràng.

Kết quả cuối cùng của giai đoạn này là các đặc điểm kỹ thuật thiết kế. Nó bao gồm các chỉ định về thiết kế kiến trúc, yêu cầu hệ thống cũng như đại diện Back-end, Front-end,… cho phép cả nhóm phát triển theo dõi toàn bộ quá trình phát triển nên phần mềm.

**Bước 3: Development (Thực hiện)**

Ở bước này, các nhà phát triển phần mềm sẽ bắt đầu viết code và triển khai các thông số thiết kế đã đưa ra ở bước 2. Cụ thể, các Front-end developer sẽ xây dựng phần giao diện của phần mềm. Các Back-end developer sẽ sử dụng các loại [ngôn ngữ lập trình](https://vtc.edu.vn/ngon-ngu-lap-trinh-la-gi/)**,** framework để lập trình trên máy chủ và cùng với các quản trị viên cơ sở dữ liệu xử lý dữ liệu.

Sau khi hoàn tất việc coding, developers sẽ deploy (triển khai) sản phẩm trong môi trường phát triển. Lập trình viên sẽ tiến hành thử nghiệm sản phẩm và có sự điều chỉnh cho phù hợp với yêu cầu đã đưa ra.

Giai đoạn này thường chiếm khá nhiều thời gian và nhân lực trong toàn bộ quy trình phát triển phần mềm.

**Bước 4: Testing (Kiểm thử phần mềm)**

Sau khi hoàn tất phần lập trình phần mềm, sản phẩm sẽ tiếp tục chuyển cho các tester (người kiểm thử phần mềm). Các tester sẽ tạo ra các tình huống kiểm thử (test case) và tiến hành kiểm thử phần mềm. Mục đích của việc kiểm thử phần mềm là xác minh và đảm bảo chất lượng của sản phẩm đúng như yêu cầu để ra.

Sau khi kiểm thử, tester sẽ cập nhật các lỗi vào công cụ quản lý và thông báo các bug (lỗi) cho developers. Bước này, tester sẽ ngồi cùng với developers để xử lý các bug hiện có và cập nhật vào hệ thống quản lý lỗi. Tùy vào mô hình phát triển phần mềm được lựa chọn ở bước 1 mà hoạt động của developer và tester có thể tiến hành lần lượt hoặc diễn ra song song.

**Bước 5: Deployment stage (Giai đoạn triển khai)**

Sau khi hoàn tất kiểm thử, phần mềm không còn lỗi, các nhà phát triển sẽ triển khai sản phẩm trên Production environment (môi trường chứa ứng dụng thật, chạy với người dùng thật, dữ liệu thật) và cung cấp sản phẩm hoàn thiện cho khách hàng.

Sau khi đăng ký, thử nghiệm Beta sẽ được tiến hành để thu thập các phản hồi của người dùng thực tế để hoàn thiện chất lượng phần mềm khi triển khai ở quy mô lớn. Ở bước này, developer cũng cần phải lên kế hoạch chuẩn bị cho mọi trường hợp bất trắc có thể xảy ra để chủ động hơn trong việc giải quyết các sự cố bất ngờ.

**Bước 6: Maintenance (Duy trì)**

Sau khi phần mềm được đưa vào vận hành chính thức, khách hàng đã bắt đầu sử dụng phần mềm ở mức chất lượng cao nhất, bước tiếp theo chúng ta cần phải bảo trì sản phẩm. Công ty sẽ thành lập một nhóm chuyên về bảo trì và quản lý các vấn để người dùng gặp phải trong quá trình sử dụng sản phẩm. Họ sẽ quản lý và giải quyết tất cả các vấn để của người dùng gặp phải. Đồng thời, phần mềm cũng được cập nhật sau khi triển khai để loại bỏ các lỗi và cải thiện hiệu suất.

**Lý do cần có quy trình phần mềm trong việc sản xuất phần mềm:**

**Tăng hiệu quả:** Quy trình phần mềm giúp tối ưu hóa quá trình phát triển, làm giảm sự lãng phí thời gian và nguồn lực. Điều này giúp tăng hiệu suất và giảm thời gian đưa sản phẩm ra thị trường.

**Đảm bảo chất lượng:** Quy trình đặt ra các bước kiểm thử và đánh giá chất lượng phần mềm, đảm bảo rằng sản phẩm cuối cùng đáp ứng được các yêu cầu về tính ổn định, an toàn và hiệu suất.

**Quản lý rủi ro:** Quy trình giúp xác định và quản lý các rủi ro trong quá trình phát triển phần mềm, cho phép dự án sẵn sàng đối phó với các vấn đề không mong muốn.

**Tạo tính nhất quán:** Quy trình định nghĩa các quy tắc và tiêu chuẩn cần phải tuân thủ trong dự án. Điều này tạo ra tính nhất quán trong việc làm việc của các thành viên trong nhóm phát triển.

**Kết hợp công việc nhóm:** Quy trình phần mềm cho phép nhiều người làm việc cùng nhau trên cùng một dự án một cách hiệu quả. Các thành viên của nhóm biết được vai trò của họ và cách họ tương tác với nhau.

**Dễ dàng kiểm soát và đối phó:** Quy trình giúp theo dõi tiến trình dự án, đưa ra các báo cáo và đánh giá tiến độ. Điều này giúp quản lý dự án dễ dàng đưa ra quyết định và thực hiện điều chỉnh khi cần thiết.

**Tạo khả năng mở rộng:** Một quy trình phần mềm tốt có thể dễ dàng được mở rộng và điều chỉnh để phản ánh sự thay đổi trong quy mô hoặc yêu cầu dự án.

**Câu 3: Vẽ & giải thích các pha (phase) trong quy trình sản xuất phần mềm thác nước (waterfall)**

A diagram of a software development process

Description automatically generated

* **Đây được coi như là mô hình phát triển phần mềm đầu tiên được sử dụng.**
* **Mô hình này áp dụng tuần tự các giai đoạn của phát triển phần mềm.**
* **Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau. Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc. Đặc biệt không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.**

**Phân tích mô hình**

* **Requirement gathering**: Thu thập và phân tích yêu cầu được ghi lại vào tài liệu đặc tả yêu cầu trong giai đoạn này.
* **System Analysis**: Phân tích thiết kế hệ thống phần mềm, xác định kiến trúc hệ thống tổng thể của phần mềm.
* **Coding**: Hệ thống được phát triển theo từng unit và được tích hợp trong giai đoạn tiếp theo. Mỗi Unit được phát triển và kiểm thử bởi dev được gọi là Unit Test.
* **Testing**: Cài đặt và kiểm thử phần mềm. Công việc chính của giai đoạn này là kiểm tra và sửa tất cả những lỗi tìm được sao cho phần mềm hoạt động chính xác và đúng theo tài liệu đặc tả yêu cầu.
* **Implementation**: Triển khai hệ thống trong môi trường khách hàng và đưa ra thị trường.
* **Operations and Maintenance**: Bảo trì hệ thống khi có bất kỳ thay đổi nào từ phía khách hàng, người sử dụng.

**Ứng dụng**

**Mô hình thường được áp dụng cho các dự án phần mềm như sau:**

* **Các dự án nhỏ , ngắn hạn.**
* **Các dự án có ít thay đổi về yêu cầu và không có những yêu cầu không rõ ràng.**

**Ưu điểm**

* **Dễ sử dụng, dễ tiếp cận, dễ quản lý.**
* **Sản phẩm phát triển theo các giai đoạn được xác định rõ ràng.**
* **Xác nhận ở từng giai đoạn, đảm bảo phát hiện sớm các lỗi.**

**Nhược điểm**

* **Ít linh hoạt, phạm vi điều chỉnh hạn chế.**
* **Rất khó để đo lường sự phát triển trong từng giai đoạn.**
* **Mô hình không thích hợp với những dự án dài, đang diễn ra, hay những dự án phức tạp, có nhiều thay đổi về yêu cầu trong vòng đời phát triển.**
* **Khó quay lại khi giai đoạn nào đó đã kết thúc.**

**Mô hình tiến hóa (Evolutionary / Prototype)**

Mô tả

* Quy trình bắt đầu từ việc thu thập yêu cầu với sự có mặt của đại diện cả phía phát triển lẫn khách hàng. Nhằm định ra mục tiêu tổng thể của hệ thống phần mềm sau này và ghi nhận các yêu cầu có thể biết được và cần làm rõ.
* Sau đó thực hiện thiết kế nhanh tập trung chuyển tải các khía cạnh thông qua nguyên mâu. Từ đó giúp khách hàng có thể hình dung, đánh giá và hoàn chỉnh yêu cầu cho toàn hệ thống. Điều này giúp tinh chỉnh yêu cầu và đội ngũ phát triển hiểu được những gì cần phát triển.
* Mô hình này thường được triển khai trong trong thời gian ngắn. Tiếp sau giai đoạn làm prototype có thể chu trình theo mô hình thác thước hoặc mô hình khác.

A diagram of a prototype model

Description automatically generated

**Phân tích**

* **Requirement:** Là giai đoạn đầu liên quan đến việc thiết lập các yêu cầu của hệ thống mong muốn.
* **Design:** Khi đã xác định các yêu cầu hệ thống mong muốn thì thiết kế ý tưởng cơ bản được hình thành.
* **Prototype formation:** Nhờ có thiết kế ý tưởng cơ bản, nguyên mẫu hoạt động được xây dựng cho hệ thống mong muốn.
* **Initial Evaluation:** Ở giai đoạn này thì mẫu được khách hàng thử nghiệm và đánh giá các chức năng cùng hạn chế.
* **Refining Prototype:** Lúc này nguyên mẫu được tinh chỉnh cùng phân tích đánh giá do khách hàng thực hiện. Sau đó quá trình tinh chỉnh được thực hiện xong thì hệ thống cuối cùng được tạo ra để sử dụng trong thời gian thực.

**Ứng dụng**

* **Mô hình được áp dụng khi yêu cầu của hệ thống rõ ràng. Các chức năng cơ bản của hệ thống mong muốn vẫn chưa được đánh giá.**
* **Khi các yêu cầu của hệ thống kết quả cần phải được thay đổi. Cần hiển thị các chức năng kỹ thuật của sản phẩm mong muốn bằng cách tạo nguyên mẫu.**
* **Thời gian phát triển bị giảm nớt và các thành phần có thể tái sử dụng.**
* **Đánh giá ban đầu được đưa ra nhanh chóng và khách hàng có thể đưa ra phản hồi theo từng nguyên mẫu.**

**Ưu điểm**

**Người dùng sẽ hình dung ra được chức năng cùng đặc điểm của hệ thống. Đồng thời cải thiện sự liên hệ giữa nhà phát triển và người sử dụng.**

**Nhược điểm**

* **Trong trường hợp các nguyên mẫu không chuyển tải hết các chức năng và đặc điểm của hệ thống phần mềm. Người dùng sẽ thất vọng và mất đi sự quan tâm đến hệ thống được phát triển.**
* **Mô hình nguyên mẫu được triển khai nhanh, vội vàng và có thể thiếu sự phân tích đánh giá cần thận các khía cạnh liên quan đến hệ thống cuối cùng.**
* **Loại mô hình này chưa thể cải thiện được việc loại trừ khoảng cách giữa yêu cầu và ứng dụng cuối cùng.**