**Nguyễn Công Minh**

**Chương 1: Mô hình truyền thống.**

## **Mô hình xoắn ốc**

### **Mô tả**

* Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.
* Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.
* Mô hình này sử dụng những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, …

### **Phân tích mô hình**

Các pha trong quy trình phát triển xoắn ốc bao gồm:

* ****Objective identification- Thiết lập mục tiêu****: xác định mục tiêu, đối tượng cho từng pha của dự án.
* ****Alternate evaluation- Đánh giá và giảm thiểu rủi ro****: đánh giá rủi ro và thực hiện các hành động để giảm thiểu rủi ro.
* ****Product development- Phát triển sản phẩm****: Lựa chọn mô hình phù hợp để phát triển hệ thống.
* ****Next phase planning- Lập kế hoạch****: đánh giá dự án và lập kế hoạch cho pha tiếp theo.

### **Ứng dụng**

Mô hình này thường được sử dụng cho các ứng dụng lớn và các hệ thống được xây dựng theo các giai đoạn nhỏ hoặc theo các phân đoạn.

### **Ưu điểm**

* Tốt cho các hệ phần mềm quy mô lớn.
* Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa.
* Đánh giá thực tế hơn như là một quy trình làm việc, bởi vì những vấn đề quan trọng đã được phát hiện sớm hơn.

### **Nhược điểm**

* Manager cần có kỹ năng tốt để quản lý dự án, đánh giá rủi ro kịp thời.
* Chi phí cao và mất nhiều thời gian để hoàn thành dự án.
* Phức tạp và không thích hợp với các dự án nhỏ và ít rủi ro.
* Yêu cầu thay đổi thường xuyên dẫn đến lặp vô hạn.
* Chưa được dùng rộng rãi.

## **1.3 Mô hình Agile**

Agile là một phương pháp phát triển phần mềm linh hoạt để làm sao đưa sản phẩm đến tay người dùng càng nhanh càng tốt và được xem như là sự cải tiến so với những mô hình cũ như mô hình “Thác nước (waterfall)” hay “CMMI”. Phương thức phát triển phần mềm Agile là một tập hợp các phương thức phát triển lặp và tăng dần trong đó các yêu cầu và giải pháp được phát triển thông qua sự liên kết cộng tác giữa các nhóm tự quản và liên chức năng.

### **Mô tả**

* Dựa trên mô hình iterative and incremental.
* Các yêu cầu và giải pháp phát triển dựa trên sự kết hợp của các function.
* Trong Agile, các tác vụ được chia thành các khung thời gian nhỏ để cung cấp các tính năng cụ thể cho bản phát hành cuối.

### **Ứng dụng**

* Có thể được sử dụng với bất kỳ loại hình dự án nào, nhưng cần sự tham gia và tính tương tác của khách hàng.
* Sử dụng khi khách hàng yêu cầu chức năng sẵn sàng trong khoảng thời gian ngắn.

### **Ưu điểm**

* Tăng cường tình thần làm việc nhóm và trao đổi công việc hiệu quả.
* Các chức năng được xây dựng nhanh chóng và rõ ràng, dế quản lý.
* Dễ dàng bổ sung, thay đổi yêu cầu.
* Quy tắc tối thiểu, tài liệu dễ hiểu, dễ sử dụng.

### **Nhược điểm**

Mô hình Agile được sử dụng rộng rãi trên thế giới nhưng cũng không đồng nghĩa với phù hợp với tất cả các dự án phần mềm.

* Không thích hợp để xử lý các phụ thuộc phức tạp.
* Có nhiều rủi ro về tính bền vững, khả năng bảo trì và khả năng mở rộng.
* Cần một team có kinh nghiệm.
* Phụ thuộc rất nhiều vào sự tương tác rõ ràng của khách hàng.
* Chuyển giao công nghệ cho các thành viên mới trong nhóm có thể khá khó khăn do thiếu tài liệu.

## **1.4 Mô hình tiếp cận lặp**

### **Mô tả**

* Một mô hình được lặp đi lặp lại từ khi start cho đến khi làm đầy đủ spec.Quá trình này sau đó được lặp lại, tạo ra một phiên bản mới của phần mềm vào cuối mỗi lần lặp của mô hình.
* Thay vì phát triển phần mềm từ spec đặc tả rồi mới bắt đầu thực thi thì mô hình này có thể review dần dần để đi đến yêu cầu cuối cùng.

### **Ứng dụng**

* Yêu cầu chính phải được xác định; tuy nhiên, một số chức năng hoặc yêu cầu cải tiến có thể phát triển theo thời gian.
* Một công nghệ mới đang được sử dụng và đang được học tập bởi nhóm phát triển trong khi làm việc trong dự án.
* Phù hợp cho các dự án lớn và nhiệm vụ quan trọng.

### **Ưu điểm**

* Xây dựng và hoàn thiện các bước sản phẩm theo từng bước.
* Thời gian làm tài liệu sẽ ít hơn so với thời gian thiết kế.
* Một số chức năng làm việc có thể được phát triển nhanh chóng và sớm trong vòng đời.
* Ít tốn kém hơn khi thay đổ phạm vi, yêu cầu.
* Dễ quản lý rủi ro.
* Trong suốt vòng đời, phần mềm được sản xuất sớm để tạo điều kiện cho khách hàng đánh giá và phản hồi.

### **Nhược điểm**

* Yếu cầu tài nguyên nhiều.
* Các vấn đề về thiết kế hoặc kiến trúc hệ thống có thể phát sinh bất cứ lúc nào.
* Yêu cầu quản lý phức tạp hơn.
* Tiến độ của dự án phụ thuộc nhiều vào giai đoạn phân tích rủi ro.

## **1.5 Mô hình tăng trưởng**

### **Mô tả**

* Spec được chia thành nhiều phần.
* Chu kỳ được chia thành các module nhỏ, dễ quản lý.
* Mỗi module sẽ đi qua các yêu cầu về thiết kế, thực hiện, … như 1 vòng đời phát triển thông thường.

### **Ứng dụng**

* Áp dụng cho những dự án có yêu cầu đã được mô tả, định nghĩa và hiểu một cách rõ ràng.
* Khahcs hàng có nhu cầu về sản phẩm sớm.

### **Ưu điểm**

* Phát triển nhanh chóng.
* Mô hình này linh hoạt hơn, ít tốn kém hơn khi thay đổi phạm vi và yêu cầu.
* Dễ dàng hơn trong việc kiểm tra và sửa lỗi.

### **Nhược điểm**

* Cần lập plan và thiết kế tốt.
* Tổng chi phí là cao hơn so với mô hình thác nước.

## **1.6 Mô hình chữ V( V model)**

### **Mô tả**

* Mô hình chữ V là một phần mở rộng của mô hình thác nước và được dựa trên sự kết hợp của một giai đoạn thử nghiệm cho từng giai đoạn phát triển tương ứng. Đây là một mô hình có tính kỷ luật cao và giai đoạn tiếp theo chỉ bắt đầu sau khi hoàn thành giai đoạn trước.
* Với V model thì công việc test được tham gia ngay từ đầu.

### **Ứng dụng**

* Yêu cầu được xác định rõ ràng.
* Xác định sản phẩm ổn định.
* Công nghệ không thay đổi và được hiểu rõ bởi nhóm dự án.
* Không có yêu cầu không rõ ràng hoặc không xác định.
* Dự án ngắn.

### **Ưu điểm**

* Đây là một mô hình có tính kỷ luật cao và các giai đoạn được hoàn thành cùng một lúc.
* Hoạt động tốt cho các dự án nhỏ, khi các yêu cầu được hiểu rất rõ.
* Đơn giản và dễ hiểu và dễ sử dụng, dễ quản lý.

### **Nhược điểm**

* Khó quản lý kiểm soát rủi ro, rủi ro cao.
* Không phải là một mô hình tốt cho các dự án phức tạp và hướng đối tượng.
* Mô hình kém cho các dự án dài và đang diễn ra.
* Không thích hợp cho các dự án có nguy cơ thay đổi yêu cầu trung bình đến cao.

## **1.7 Mô hình Scrum**

### **Mô tả**

* Chia các yêu cầu ra làm theo từng giai đoạn. Mỗi 1 giai đoạn(sprint) chỉ làm 1 số lượng yêu cầu nhất định.
* Mỗi một sprint thường kéo dài từ 1 tuần đến 4 tuần ( ko dài hơn 1 tháng).
* Đầu sprint sẽ lên kế hoạch làm những yêu cầu nào. Sau đó, sẽ thực hiện code và test. Cuối sprint là 1 sản phẩm hoàn thiện cả code lẫn test có thể demo và chạy được.
* Hoàn thành sprint 1, tiếp tục làm sprint 2, sprint... cho đến khi hoàn thành hết các yêu cầu.
* Trong mỗi 1 sprint thì sẽ có họp hàng ngày – daily meeting từ 15 – 20 phút. Mỗi thành viên sẽ báo cáo: Hôm qua tôi đã làm gì? Hôm nay tôi sẽ làm gì? Có gặp khó khăn gì không?
* Scrum là mô hình hướng khách hàng (Customer oriented).

### **Các nhân tố tạo nên quy trình Scrum**

Có 3 thành tố quan trọng cấu thành nên SCRUM:

* ****Tổ chức (Organization)****
  + Tổ chức nhóm dự án và Roles: Vài trò.
  + Product Owner: Người sở hữu sản phẩm.
  + ScrumMaster: Người điều phối.
  + Development Team: Nhóm phát triển.
* ****Tài liệu (Atifacts):**** đó chính là các kết quả đầu ra.
  + Product Backlog: Danh sách các chức năng cần phát triển của sản phẩm.
  + Sprint Backlog: Danh sách các chức năng cần phát triển cho mỗi giai đoạn.
  + Estimation:Kết quả ước lượng của team.
* ****Qui trình(Process)****: Qui định cách thức vận hành của SCRUM.
  + Sprint Planning meeting: Hoạch định cho mỗi giai đoạn.
  + Review: Tổng kết cho mỗi giai đoạn.
  + Daily Scrum Meeting: Review hàng ngày.

### **Tổ chức dự án**

* ****Product Owner****
  + Product Owner là người sở hữu sản phẩm, người quyết định sản phẩm có những chức năng nào và là người quyết định Product Backlog.
  + Thông thường Role này được khách hàng hoặc người đại diện cho khách hàng đảm nhận.
* ****ScrumMaster****
  + Scrum Master là người đảm bảo các qui trình của Scrum được thực hiện đúng và thuận lợi.
* ****Development Team****
  + Một nhóm từ 4-7 kỹ sư phần mềm chịu trách nhiệm phát triển sản phẩm.
  + Nhóm dự án phải làm việc với Product Owner để quyết định những gì sẽ làm trong Sprint (giai đoạn )này và kết quả sẽ ra sao.
  + Thảo luận để đưa ra các giải pháp, ước lượng thời gian thực hiện công việc, họp đánh giá kết quả công việc.
* ****Product Backlog****
  + Product Backlog là danh sách các chức năng cần được phát triển của sản phẩm.
  + Danh sách này do Product Owner quyết định.
  + Thường xuyên được cập nhật để đáp ứng được nhu cầu thay đổi của khách hàng và dự án.

### **Ưu điểm**

* Một người có thể thực hiện nhiều việc ví dụ như dev có thể test.
* Phát hiện lỗi sớm.
* Có khả năng áp dụng được cho những dự án mà yêu cầu khách hàng không rõ ràng ngay từ đầu.

### **Nhược điểm**

* Trình độ của nhóm cần có một kỹ năng nhất định.
* Phải có sự hiểu biết về mô hình aglie.
* Khó khăn trong việc xác định ngân sách và thời gian.
* Luôn nghe ý kiến phản hồi từ khách hàng và thay đổi theo nên thời gian sẽ kéo dài.
* Vai trò của PO rất quan trọng, PO là người định hướng sản phẩm. Nếu PO làm không tốt sẽ ảnh hưởng đến kết quả chung.

## **1.8 Mô hình RAD**

### **Mô tả**

* Mô hình RAD là một phương pháp phát triển phần mềm sử dụng quy hoạch tối thiểu có lợi cho việc tạo mẫu nhanh.
* Các mô-đun chức năng được phát triển song song như nguyên mẫu và được tích hợp để tạo ra sản phẩm hoàn chỉnh để phân phối sản phẩm nhanh hơn.
* Đảm bảo rằng các nguyên mẫu được phát triển có thể tái sử dụng được.

### **Ứng dụng**

Mô hình RAD có thể được áp dụng thành công cho các dự án:

* Module hóa rõ ràng. Nếu dự án không thể được chia thành các mô-đun, RAD có thể không thành công.
* RAD nên được sử dụng khi có nhu cầu để tạo ra một hệ thống có yêu cầu khách hàng thay đổi trong khoảng thời gian nhỏ 2-3 tháng.
* Nên được sử dụng khi đã có sẵn designer cho model và chi phí cao.

### **Ưu điểm**

* Giảm thời gian phát triển.
* Tăng khả năng tái sử dụng của các thành phần.
* Đưa ra đánh giá ban đầu nhanh chóng.
* Khuyến khích khách hàng đưa ra phản hồi.

### **Nhược điểm**

* Trình độ của nhóm cần có một kỹ năng nhất định.
* Chỉ những hệ thống có module mới sử dụng được mô hình này.

# **2. Tổng kết vòng đời phát triển phần mềm**

* SDLC( Software Development Life Circicle) là một quá trình đi theo suốt trong một dự án phần mềm, trong một tổ chức phần mềm. Nó bao gồm một kế hoạch chi tiết mô tả cách phát triển, duy trì, thay thế và thay đổi hoặc nâng cao phần mềm cụ thể. Vòng đời xác định một phương pháp để cải thiện chất lượng phần mềm và quy trình phát triển tổng thể.
* Mỗi mô hình sẽ có sự khác nhau, tuy nhiên, mỗi mô hình bao gồm tất cả hoặc một số giai đoạn / hoạt động / nhiệm vụ sau.

### **Planning - Lập kế hoạch**

* Phân tích yêu cầu được thực hiện bởi các senior member của nhóm với đầu vào từ khách hàng, bộ phận bán hàng, khảo sát thị trường và các chuyên gia trong ngành.
* Lập kế hoạch cho các yêu cầu đảm bảo chất lượng và xác định các rủi ro liên quan đến dự án.

### **Defining - Xác định yêu cầu**

* Xác định rõ ràng và ghi lại các yêu cầu.
* Thực hiện xác định yêu cầu thông qua một tài liệu SRS (Software Requirement Specification) bao gồm tất cả các yêu cầu sản phẩm được thiết kế và phát triển trong suốt vòng đời của dự án.

### **Designing - Phân tích thiết kế kiến trúc hệ thống**

* Trong giai đoạn này, thiết kế hệ thống và phần mềm được chuẩn bị từ các đặc tả yêu cầu đã được nghiên cứu trong giai đoạn đầu tiên.
* Thiết kế hệ thống giúp xác định các yêu cầu phần cứng và kiến trúc hệ thống tổng thể.

### **Building - Phát triển**

* Khi nhận được tài liệu thiết kế hệ thống, công việc được chia thành các module nhỏ và coding được bắt đầu.
* Đây là giai đoạn trọng tâm và dài nhất của vòng đời phát triển phần mềm.

### **Testing - Kiếm thử**

* Sau khi coding được phát triển, nó được kiểm tra dựa trên các yêu cầu để đảm bảo rằng sản phẩm thực sự hoạt động đúng trong giai đoạn phân tích yêu cầu.
* Trong giai đoạn này tất cả các loại kiểm thử chức năng như kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống, kiểm thử chấp nhận được thực hiện cũng như test phi chức năng cũng được thực hiện.

### **Deployment - Phát hành/triển khai**

* Sau khi thử nghiệm thành công, sản phẩm được phân phối / triển khai cho khách hàng để họ sử dụng.
* Khách hàng sẽ thực hiện test beta. Nếu có bất kỳ vấn đề nào xảy ra, thì họ sẽ báo cáo cho nhóm kỹ thuật để được hỗ trợ.

### **Maintenance - Bảo trì**

* Trong quá trình bảo trì của SDLC, hệ thống được đánh giá để đảm bảo nó không trở nên lỗi thời.
* Một khi khách hàng bắt đầu sử dụng hệ thống đã phát triển thì những vấn đề thực sự xuất hiện và cần được giải quyết theo thời gian.  
  Như vậy trong bài chia sẻ này, mình đã giới thiệu ngắn gọn một số mô hình phát triển phần mềm phổ biến hiện nay. Hy vọng rằng kiến thức mình tổng hợp sẽ giúp ích cho các bạn trong quá trình học tập cũng như làm việc.

**Chương 2: Áp dụng mô hình scrum vào dự án**

Có 4 bước lập kế hoạch dự án mọi nhà quản lý cần ghi nhớ trong dự án chúng ta:

***Project Goal - xác định mục tiêu dự án***

Sự thành công của một dự án phụ thuộc vào việc dự án đó có làm thỏa mãn mong muốn của những bên liên quan (stakeholders) hay không. Tại bước đầu tiên này, điều quan trọng là các nhà quản lý cần xác định chính xác các bên liên quan là ai và từ đó tìm ra những nhu cầu của họ.

Các bên liên quan có thể là:

Nhà tài trợ

Nhóm dự án tạo ra sản phẩm cuối cùng. Họ tham gia vào nhiều khía cạnh quan trọng như phát triển, thiết kế, v.v. nhưng  không phê duyệt dự án

Các end-user

Các nhà phân tích rủi ro, chuyên gia mua hàng, v.v.

Khi đã biết các bên liên quan là ai, nhà quản lý cần tìm hiểu nhu cầu của họ và từ đó thiết lập ra các mục tiêu. Các cách để có thể khai thác thông tin từ các bên liên quan có thể là tổ chức cuộc họp, gặp gỡ 1-1 hoặc thực hiện các cuộc khảo sát.

Một dự án có thể có nhiều mục tiêu nhưng hãy đảm bảo rằng các mục tiêu tuân theo quy chuẩn [SMART](https://en.wikipedia.org/wiki/SMART_criteria?utm_content=resources.base.vn/productivity/4-buoc-len-ke-hoach-quan-ly-du-an-549" \t "https://resources.base.vn/productivity/_blank)(Specific - cụ thể; Measurable - có thể đo lường; Attainable - có thể đạt được; Realistic - thực tế; Timely - có thời gian cụ thể). Sau khi hoàn thành thiết lập mục tiêu, điều cần làm tiếp theo là xác định thời gian chuyển giao..

***Project deliverable - xác định thời gian chuyển giao***

Trong quản lý dự án, có những mốc thời gian ghi nhận sự giao nhận/ cung cấp sản phẩm, dịch vụ giữa khách hàng và công ty. Những mốc thời gian này được gọi là Deliverable. Việc chuyển giao sản phẩm có thể được coi là những dự án nhỏ trong dự án lớn.

Sự giao nhận thường cụ thể và có thể đo lường. Đối tượng của sự giao nhận có thể là khách hàng bên ngoài hoặc các bên liên quan trong dự án. Sản phẩm giao nhận cũng rất đa dạng. Đó có thể là phần mềm, tài liệu thiết kế, chương trình đào tạo hoặc các tài sản khác được yêu cầu trong kế hoạch dự án.

Để có thể quản lý thời gian chuyển giao hiệu quả, nhà quản lý có thể áp dụng những lời khuyên sau:

Xác định sản phẩm bàn giao và các yêu cầu của các bên liên quan

Đảm bảo các yêu cầu đáp ứng mô hình SMART

Phân chia các sản phẩm bàn giao thành các giai đoạn để có thể theo dõi dễ dàng hơn

Xác định các số liệu đo lường mức độ chấp nhận của mỗi lần bàn giao (VD: số lượng, chất lượng, tình chất sản phẩm,...). Điều này sẽ giúp tránh những thay đổi trong việc cung cấp sau này

Xác định rõ thời gian bàn giao và nên gắn thời gian bàn giao với các mốc quan trọng của dự án để đảm bảo theo dõi tốt hơn

Sử dụng các phần mềm quản lý dự án như Base Wework để theo dõi và quản lý việc phân phối dự án dễ dàng hơn.

***Project Schedule - xác định lộ trình dự án***

Các mốc thời gian trong dự án nên có nhiều hơn ngày bắt đầu và ngày kết thúc. Nhà quản lý cần lên lịch cho từng phần của dự án từ giai đoạn triển khai, giai đoạn phát triển và thậm chí cả những giai đoạn nhỏ ở giữa. Lộ trình càng chi tiết, nhà quản lý càng dễ dàng theo dõi tiến độ của nhóm mình quản lý và đồng thời các bên liên quan cũng có thể chủ động nắm bắt được lộ trình chung.

Vậy, làm thế nào để tạo được một lộ trình dự án?

Lộ trình dự án được tạo lập qua 4 bước:

Xác định nhiệm vụ cần làm

Xác định độ phụ thuộc giữa các nhiệm vụ

Xác định những tài nguyên cần có để làm nhiệm vụ

Xác định thời gian hoàn thành

Sau khi có thời gian chuyển giao, nhà quản lý cần liệt kê cụ thể những công việc cần làm để có thể hoàn thành thời gian chuyển giao đúng hạn. Tiếp theo, hãy xác định sự phụ thuộc giữa các công việc nhằm xác định xem công việc nào cần được thực hiện trước tiên. Bằng cách đó, nhà quản lý có thể hoàn thành trước các công việc quan trọng.

Đừng quên xem xét các mức phụ thuộc khác nhau giữa các công việc bởi chúng có thể ảnh hưởng đến ngày bắt đầu và kết thúc của mỗi nhiệm vụ và thậm chí còn có thể ảnh hưởng đến toàn bộ dự án. Những mức quan hệ phụ thuộc giữa các nhiệm vụ có thể kể đến như: Finish-to-start - nhiệm vụ A phải được hoàn thành trước khi nhiệm vụ B bắt đầu - hoặc Start-to-start - nhiệm vụ A phải được bắt đầu trước khi nhiệm vụ B có thể bắt đầu.

Khi các đề mục công việc được sắp xếp hợp lý, nhà quản lý sẽ xác định những tài nguyên cần sử dụng. Điều này thật sự quan trọng vì rất có thể nhiều tài nguyên sẽ được sử dụng song song, gây nên sự xung đột lịch trình. Để giải quyết vấn đề này, hãy xem xét kỹ càng sự sẵn có của tài nguyên và sắp xếp lịch trình sử dụng theo mức độ ưu tiên.

Sau khi đã sắp xếp tài nguyên đối với từng nhiệm vụ, tiếp theo, hãy xác định xem mỗi nhiệm vụ cần bao nhiêu thời gian để hoàn thành. Thời gian hoàn thành nhiệm vụ có thể tính bằng giờ, ngày, tuần hoặc thậm chí vài tháng tùy thuộc vào quy mô và độ phức tạp của dự án.

Có một lỗi quản lý rất phổ biến đó là không tính đủ thời gian để hoàn thành một nhiệm vụ. Khi trường hợp này xảy ra, tất cả các bên liên quan phải vội vã hoàn thành nhiệm vụ và điều này dẫn đến chất lượng công việc bị giảm sút.

Để có thể lên kế hoạch cho từng đầu việc hiệu quả, nhà quản lý có thể tham khảo những dự án có đầu mục công việc mang đặc điểm tương tự. Bên cạnh đó, những vấn đề sau cũng cần được lưu ý khi lên kế hoạch thời gian cho từng nhiệm vụ:

Lịch trình cá nhân

Ngày nghỉ

Mức độ hữu dụng của từng tài nguyên

Sự chậm trễ do các vấn đề khác

***Supporting plans - kế hoạch hỗ trợ***

Một kế hoạch quản lý dự án tốt không thể thiếu các kế hoạch hỗ trợ cũng như các phương án quản lý rủi ro. Những kế hoạch hỗ trợ có thể là: Kế hoạch nhân sự, Kế hoạch thông báo và Kế hoạch quản trị rủi ro

*Kế hoạch nhân sự*

Hãy xác định các cá nhân và tổ chức có vai trò quan trọng trong dự án. Đối với mỗi cá nhân và tổ chức đó, hãy mô tả vai trò và trách nhiệm của họ đối với toàn bộ dự án, sau đó chỉ định số lượng và những cá nhân cụ thể đối với từng nhiệm vụ.

*Kế hoạch thông báo*

Hãy soạn thảo một văn bản để phổ biến cho các bên liên quan nắm được ai sẽ liên tục được thông báo về tiến độ dự án và cách họ sẽ nhận được thông tin. Cách phổ biến nhất để cập nhật tình hình là tạo báo cáo hàng tuần, hàng tháng, mô tả cách thức thực hiện dự án hay các mốc công việc quan trọng đã đạt được,..

*Kế hoạch quản trị rủi ro*

Quản lý rủi ro là một phần quan trọng của quản lý dự án, tuy nhiên, kế hoạch này thường bị bỏ qua. Càng dự đoán được nhiều rủi ro cho dự án, nhà quản lý càng có nhiều kế hoạch dự phòng và chuẩn bị tốt cách đối phó khi có tình huống xấu xảy ra.

Dưới đây là một số ví dụ về rủi ro trong các dự án:

Dự tính sai thời gian và chi phí

Đánh giá của khách hàng và tốc độ phản hồi chậm

Ngân sách bị cắt giảm đột ngột

Không phân định rõ vai trò và trách nhiệm của các bên

Một trong các bên thay đổi yêu cầu khi dự án đã bắt đầu

Một trong các bên phát sinh nhu cầu mới sau khi dự án đã bắt đầu

Giao tiếp không hiệu quả giữa các bên

Rủi ro liên quan đến tài nguyên

Nhà quản lý có thể theo dõi rủi ro qua các bản ghi chép. Các bản ghi sẽ liệt kê từng rủi ro và những hành động cụ thể cần làm trong trường hợp rủi ro phát sinh. Nhà quản lý nên thường xuyên xem bản ghi chép, thêm mới những rủi ro xảy ra trong vòng đời dự án và cách thức giải quyết. Hãy nhớ rằng rủi ro sẽ không thể biến mất nếu nhà quản lý không có bất kỳ hành động nhằm giải quyết và ngăn chặn. Kết quả là toàn bộ dự án sẽ bị chậm tiến độ hoặc thậm chí thất bại.

**Đừng quên rằng việc lập kế hoạch và triển khai, theo dõi tiến độ dự án đều có thực hiện dễ dàng bằng một phần mềm quản lý.**Sáng suốt trong việc chọn lựa công cụ hỗ trợ chính là điểm mấu chốt giúp nhà quản lý tăng doanh thu, giảm chi phí và vận hành mọi thứ đơn giản hơn nhiều lần.

Dưới đây là giao diện Gantt chart của Base Wework, phần mềm quản lý công việc và dự án được cung cấp bởi Base.vn - với lượng khách hàng doanh nghiệp đã vượt mức 5000 vào năm 2020.