

Mục lục

I. Xác định Rủi ro Dự án	2
1. Các loại rủi ro	2
1. Rủi ro Kỹ thuật:.....	2
2. Rủi ro Quản lý:.....	2
3. Rủi ro Tài chính:	3
4. Rủi ro Hợp đồng và Pháp lý:	3
5. Rủi ro Chấp nhận của Người dùng:	3
6. Rủi ro Bảo trì:	3
2. Check List.....	3
1. Thiếu lãnh đạo (Lack of leadership)	3
2. Yêu cầu không rõ ràng (Unclear requirements).....	4
3. Thay đổi yêu cầu quá nhiều (Too many requirement changes)	4
4. Lịch trình không thực tế (Unrealistic schedules).....	5
5. Làm việc trên công nghệ mới (Working on new technology)	5
6. Làm việc trên quy trình mới (Working on new process)	6
7. Mất mát nhân lực (Manpower attrition).....	6
8. Tiến độ chậm (Slow performance).....	7
9. Điều chỉnh giữa chừng (The mid-course correction).....	7
10. Ước tính dư thừa tạo ra sự không tin tưởng (Padded estimates generate distrust)	8
11. Lời tiên tri tự hoàn thành (Self-fulfilling prophecy)	8
12. Làm việc ngược từ thời hạn (Working backward from a deadline)	8
13. Không hiểu rõ các nhiệm vụ trước (Misunderstood predecessors).....	9

14.	Lạm dụng lặp đi lặp lại (Iteration abuse).....	9
15.	Phát hiện vấn đề quá muộn (Problems are found too late).....	10
16.	Các cuộc họp lớn, vô nghĩa (Big, useless meetings).....	10
17.	"Người hùng" không thể thay thế (The indispensable "hero").....	11
II.	Phân tích Rủi ro Dự án	11
1.	Xác định độ ưu tiên cho từng rủi ro	11
III.	Đề xuất và Chọn Lựa Phản Ứng Rủi ro	12
IV.	Tạo Kế Hoạch Quản Lý Rủi ro.....	12
1.	Risk 1:.....	12
V.	Theo dõi và Giám sát Rủi ro Dự án.....	14

I. Xác định Rủi ro Dự án

1. Các loại rủi ro

1. Rủi ro Kỹ thuật:

Khả năng xảy ra lỗi phần mềm do các vấn đề liên quan đến thiết kế, triển khai, hoặc tích hợp có thể là một rủi ro kỹ thuật. Các vấn đề về hiệu suất, tương thích, và bảo mật cũng có thể được coi là rủi ro kỹ thuật.

2. Rủi ro Quản lý:

Rủi ro quản lý có thể bao gồm việc không hiểu rõ yêu cầu của dự án, thiếu tài nguyên, quản lý dự án không hiệu quả, hoặc không thống nhất trong quá trình điều phối công việc giữa các thành viên trong nhóm dự án.

3. Rủi ro Tài chính:

Các vấn đề tài chính như vượt quá ngân sách, ước lượng chi phí không chính xác, hoặc thay đổi yêu cầu có thể tạo ra rủi ro tài chính đối với dự án phần mềm.

4. Rủi ro Hợp đồng và Pháp lý:

Các rủi ro liên quan đến việc không đạt được thỏa thuận hợp đồng, xung đột về quyền sở hữu trí tuệ, hoặc rủi ro pháp lý có thể ảnh hưởng đến quá trình triển khai và sự thành công của dự án.

5. Rủi ro Chấp nhận của Người dùng:

Nếu người dùng không chấp nhận sản phẩm cuối cùng do không đáp ứng đúng yêu cầu hoặc khả năng sử dụng không hiệu quả, đó có thể coi là một rủi ro chấp nhận của người dùng.

6. Rủi ro Bảo trì:

Liên quan đến khả năng bảo trì và nâng cấp phần mềm sau khi triển khai. Nếu hệ thống không dễ bảo trì hoặc không linh hoạt, đó có thể tạo ra rủi ro bảo trì.

2. Check List

1. Thiếu lãnh đạo (Lack of leadership)

Một người quản lý dự án hiệu quả cần có các kỹ năng lãnh đạo mạnh mẽ. Họ cần có khả năng cung cấp định hướng rõ ràng, thiết lập kỳ vọng thực tế và động viên nhóm. Khi thiếu lãnh đạo, dự án có thể bị trật bánh vì các thành viên trong nhóm không biết phải làm gì hoặc không có động lực để hoàn thành công việc.

Ví dụ:

Một người quản lý dự án thiếu lãnh đạo có thể không cung cấp cho nhóm các mục tiêu rõ ràng hoặc thời hạn cụ thể. Điều này có thể khiến các thành viên trong nhóm bối rối và không biết phải bắt đầu từ đâu.

Ngoài ra, người quản lý dự án thiếu lãnh đạo có thể không cung cấp phản hồi hoặc hỗ trợ cho nhóm. Điều này có thể khiến các thành viên trong nhóm cảm thấy bị bỏ rơi và không được đánh giá cao.

2. Yêu cầu không rõ ràng (Unclear requirements)

Yêu cầu không rõ ràng có thể dẫn đến nhiều vấn đề trong dự án, bao gồm:

- Khó ước tính phạm vi công việc và ngân sách.
- Quá trình mở rộng phạm vi.
- Bỏ lỡ thời hạn.
- Chi tiêu quá mức.

Ví dụ:

Nếu yêu cầu không rõ ràng về tính năng của sản phẩm, người quản lý dự án sẽ khó ước tính bao nhiêu thời gian và tài nguyên cần thiết để phát triển sản phẩm. Điều này có thể dẫn đến thời hạn bị trễ và chi phí vượt quá ngân sách.

Ngoài ra, yêu cầu không rõ ràng có thể dẫn đến các thành viên trong nhóm tranh cãi về cách giải thích yêu cầu. Điều này có thể dẫn đến sự chậm trễ và lãng phí thời gian.

3. Thay đổi yêu cầu quá nhiều (Too many requirement changes)

Thay đổi yêu cầu là một phần bình thường của bất kỳ dự án nào. Tuy nhiên, thay đổi yêu cầu quá nhiều có thể dẫn đến nhiều vấn đề, bao gồm:

- Khó theo dõi tiến độ.
- Bỏ lỡ thời hạn.
- Chi tiêu quá mức.
- Giảm chất lượng.

Ví dụ:

Nếu các yêu cầu của sản phẩm thay đổi thường xuyên, người quản lý dự án sẽ khó theo dõi tiến độ của dự án. Điều này có thể dẫn đến việc các nhiệm vụ không được hoàn thành đúng thời hạn hoặc không đúng chất lượng.

Ngoài ra, thay đổi yêu cầu có thể khiến các thành viên trong nhóm cảm thấy thất vọng và mất động lực.

4. Lịch trình không thực tế (Unrealistic schedules)

Lịch trình không thực tế có thể dẫn đến nhiều vấn đề, bao gồm:

- Bỏ lỡ thời hạn.
- Tăng căng thẳng cho nhóm.
- Giảm chất lượng.

Ví dụ:

- Nếu lịch trình dự án không tính đến các sự chậm trễ không lường trước được, dự án có thể bị trễ. Điều này có thể dẫn đến căng thẳng cho nhóm và giảm chất lượng của sản phẩm.
- Ngoài ra, lịch trình không thực tế có thể khiến các thành viên trong nhóm cảm thấy bị áp lực và không có động lực.

5. Làm việc trên công nghệ mới (Working on new technology)

Làm việc trên công nghệ mới có thể mang lại nhiều lợi thế, nhưng cũng có thể tiềm ẩn nhiều rủi ro. Nếu không được quản lý cẩn thận, việc làm việc trên công nghệ mới có thể dẫn đến các vấn đề sau:

- Tăng thời gian và chi phí phát triển.
- Giảm chất lượng.
- Bỏ lỡ thời hạn.

Ví dụ:

Một công ty phát triển phần mềm có thể quyết định sử dụng một công nghệ mới để phát triển một sản phẩm mới. Tuy nhiên, công ty không có kinh nghiệm với công nghệ mới này. Điều này có thể dẫn đến việc phát triển sản phẩm bị chậm trễ và chi phí vượt quá ngân sách.

6. Làm việc trên quy trình mới (Working on new process)

Làm việc trên quy trình mới cũng có thể tiềm ẩn nhiều rủi ro. Nếu không được quản lý cẩn thận, việc làm việc trên quy trình mới có thể dẫn đến các vấn đề sau:

- Giảm hiệu quả.
- Tăng thời gian và chi phí phát triển.
- Bỏ lỡ thời hạn.

Ví dụ:

Một công ty sản xuất có thể quyết định áp dụng một quy trình sản xuất mới. Tuy nhiên, công ty không đào tạo đầy đủ cho nhân viên về quy trình mới này. Điều này có thể dẫn đến việc sản xuất bị chậm trễ và chất lượng sản phẩm giảm sút.

7. Mất mát nhân lực (Manpower attrition)

Mất mát nhân lực là tình trạng các thành viên trong nhóm rời khỏi dự án do các lý do khác nhau, chẳng hạn như nghỉ việc, chuyển công tác, hoặc bị sa thải. Mất mát nhân lực có thể dẫn đến các vấn đề sau:

- Giảm năng suất.
- Tăng chi phí.
- Bỏ lỡ thời hạn.
- Giảm chất lượng.

Ví dụ:

Một dự án phát triển phần mềm có thể mất một số lập trình viên quan trọng do họ được tuyển dụng bởi một công ty khác. Điều này có thể dẫn đến việc phát triển phần mềm bị chậm trễ và chi phí vượt quá ngân sách.

8. Tiến độ chậm (Slow performance)

Tiến độ chậm là tình trạng dự án không hoàn thành đúng thời hạn. Tiến độ chậm có thể do nhiều nguyên nhân, chẳng hạn như:

- Yêu cầu không rõ ràng.
- Thay đổi yêu cầu quá nhiều.
- Lịch trình không thực tế.
- Mất mát nhân lực.
- Các vấn đề kỹ thuật.

Ví dụ:

Một dự án xây dựng có thể bị chậm trễ do các vấn đề về thủ tục hành chính.

9. Điều chỉnh giữa chừng (The mid-course correction)

Điều chỉnh giữa chừng là việc thay đổi kế hoạch của dự án sau khi dự án đã bắt đầu. Điều chỉnh giữa chừng có thể cần thiết trong một số trường hợp, chẳng hạn như:

- Khi có những thay đổi về yêu cầu.
- Khi có những vấn đề không lường trước được.
- Tuy nhiên, điều chỉnh giữa chừng cũng có thể gây ra các vấn đề, chẳng hạn như:
 - Tăng chi phí.
 - Giảm chất lượng.
 - Bỏ lỡ thời hạn.

Ví dụ:

Một dự án phát triển phần mềm có thể cần điều chỉnh giữa chừng để bổ sung các tính năng mới. Điều này có thể dẫn đến việc phát triển phần mềm bị chậm trễ và chi phí vượt quá ngân sách.

10. Ước tính dư thừa tạo ra sự không tin tưởng (Padded estimates generate distrust)

Ước tính dư thừa là việc ước tính thời gian và tài nguyên cần thiết để hoàn thành các nhiệm vụ một cách quá cao. Ước tính dư thừa có thể được thực hiện để đảm bảo dự án hoàn thành đúng thời hạn và ngân sách. Tuy nhiên, ước tính dư thừa cũng có thể tạo ra sự không tin tưởng giữa người quản lý dự án và các thành viên trong nhóm.

Ví dụ:

Một người quản lý dự án có thể ước tính rằng một nhiệm vụ sẽ mất 10 ngày để hoàn thành. Tuy nhiên, thực tế nhiệm vụ chỉ mất 5 ngày để hoàn thành. Điều này có thể khiến các thành viên trong nhóm cảm thấy rằng người quản lý dự án không đáng tin cậy.

11. Lời tiên tri tự hoàn thành (Self-fulfilling prophecy)

Lời tiên tri tự hoàn thành là một hiện tượng tâm lý trong đó niềm tin của một người về một kết quả nhất định có thể dẫn đến kết quả đó. Trong bối cảnh quản lý dự án, lời tiên tri tự hoàn thành có thể xảy ra khi người quản lý dự án tin rằng dự án sẽ thất bại. Tin tưởng này có thể dẫn đến các hành động của người quản lý dự án khiến dự án thất bại.

Ví dụ:

Một người quản lý dự án tin rằng dự án sẽ vượt ngân sách. Tin tưởng này có thể dẫn đến việc người quản lý dự án cắt giảm chi phí, dẫn đến chất lượng sản phẩm giảm sút.

12. Làm việc ngược từ thời hạn (Working backward from a deadline)

Làm việc ngược từ thời hạn là việc bắt đầu dự án từ thời hạn cuối cùng và làm việc ngược lại để xác định thời hạn bắt đầu và thời hạn cho các nhiệm vụ. Làm việc ngược từ thời hạn có thể hữu ích trong một số trường hợp, chẳng hạn như:

- Khi thời hạn cuối cùng là cố định.
- Khi các yêu cầu của dự án rõ ràng.

- Tuy nhiên, làm việc ngược từ thời hạn cũng có thể dẫn đến các vấn đề, chẳng hạn như:
- Bỏ qua các yếu tố quan trọng, chẳng hạn như phạm vi dự án và các nguồn lực cần thiết.
- Tạo ra lịch trình không thực tế.

Ví dụ:

Một người quản lý dự án có thể làm việc ngược từ thời hạn cuối cùng là 1 năm để xác định thời hạn bắt đầu và thời hạn cho các nhiệm vụ. Tuy nhiên, nếu các yêu cầu của dự án không rõ ràng, người quản lý dự án có thể tạo ra một lịch trình không thực tế.

13. Không hiểu rõ các nhiệm vụ trước (Misunderstood predecessors)

Các nhiệm vụ trước là những nhiệm vụ cần phải hoàn thành trước khi một nhiệm vụ khác có thể được bắt đầu. Nếu các nhiệm vụ trước không được hiểu rõ, nó có thể dẫn đến các vấn đề sau:

- Các nhiệm vụ không được hoàn thành đúng thời hạn.
- Các nhiệm vụ được hoàn thành không đúng cách.
- Tăng chi phí.

Ví dụ:

Trong một dự án phát triển phần mềm, nhiệm vụ "Thiết kế giao diện người dùng" phải được hoàn thành trước khi nhiệm vụ "Phát triển giao diện người dùng" có thể được bắt đầu. Nếu các nhiệm vụ trước không được hiểu rõ, nó có thể dẫn đến việc giao diện người dùng được phát triển không đúng cách hoặc không phù hợp với nhu cầu của người dùng.

14. Lạm dụng lặp đi lặp lại (Iteration abuse)

Lặp đi lặp lại là một kỹ thuật project management hiệu quả có thể giúp cải thiện chất lượng sản phẩm. Tuy nhiên, lạm dụng lặp đi lặp lại có thể dẫn đến các vấn đề sau:

- Tăng chi phí.
- Tăng thời gian.

- Giảm năng suất.

Ví dụ:

Một công ty phát triển phần mềm có thể quyết định sử dụng lặp đi lặp lại để phát triển một sản phẩm mới. Tuy nhiên, công ty không có kế hoạch rõ ràng cho việc lặp đi lặp lại. Điều này có thể dẫn đến việc lặp đi lặp lại diễn ra không cần thiết và tốn kém.

15. Phát hiện vấn đề quá muộn (Problems are found too late)

Nếu các vấn đề được phát hiện quá muộn trong quá trình phát triển dự án, nó có thể dẫn đến các vấn đề sau:

- Tăng chi phí.
- Tăng thời gian.
- Giảm chất lượng.

Ví dụ:

Trong một dự án xây dựng, vấn đề về kết cấu của tòa nhà có thể được phát hiện quá muộn. Điều này có thể dẫn đến việc phải phá dỡ và xây dựng lại một phần của tòa nhà, gây tốn kém và trì hoãn tiến độ dự án.

16. Các cuộc họp lớn, vô nghĩa (Big, useless meetings)

Các cuộc họp là một phần quan trọng của bất kỳ dự án nào. Tuy nhiên, các cuộc họp lớn, vô nghĩa có thể dẫn đến các vấn đề sau:

- Tốn thời gian và tiền bạc.
- Giảm năng suất.
- Giảm động lực của nhóm.

Ví dụ:

Một công ty có thể tổ chức một cuộc họp hàng tuần với tất cả nhân viên của công ty. Tuy nhiên, cuộc họp này không có mục đích rõ ràng hoặc không có kết quả cụ thể. Điều này có thể dẫn đến việc nhân viên lãng phí thời gian và không tập trung vào công việc của mình.

17. "Người hùng" không thể thay thế (The indispensable "hero")

Khi một dự án phụ thuộc vào một cá nhân duy nhất, nó có thể dẫn đến các vấn đề sau:

- Tăng rủi ro.
- Giảm hiệu quả.
- Tăng căng thẳng cho cá nhân đó.

Ví dụ:

Trong một dự án phát triển phần mềm, một lập trình viên duy nhất là người duy nhất có thể giải quyết một vấn đề kỹ thuật phức tạp. Nếu lập trình viên đó bị ốm hoặc nghỉ việc, dự án có thể bị trì hoãn hoặc thất bại.

II. Phân tích Rủi ro Dự án

1. Xác định độ ưu tiên cho từng rủi ro

Khả năng xuất hiện

- 6 – Thường xuyên
- 4 – Hay xảy ra
- 2 – Đôi khi
- 1 – Hiếm khi

Mức độ tác động

- 8 – Trầm trọng: Có khả năng rất cao làm dự án thất bại

- 6 - Quan trọng: Gây khó khăn và làm dự án không đạt mục tiêu
- 2 – Vừa phải: Gây khó khăn cho dự án, ảnh hưởng tới việc đạt mục tiêu
- 1 – Không đáng kể: Gây khó khăn không đáng kể

Thời điểm xuất hiện

- 6 – Ngay lập tức: Rủi ro xuất hiện gần như tức khắc
- 4 – Rất gần, rủi ro sẽ xuất hiện tại một thời điểm rất gần với dự án
- 2 – Sắp xảy ra: Rủi ro sẽ xuất hiện trong tương lai gần
- 1 -Rất lâu: Rủi ro xuất hiện trong tương lai xa, chưa xác định được

III. Đề xuất và Chọn Lựa Phản Ứng Rủi ro

Các loại phản ứng trước rủi ro:

- Chấp nhận
- Tránh
- Chuyển giao
- Giảm nhẹ

IV. Tạo Kế Hoạch Quản Lý Rủi ro

1. Risk 1:

Mục tiêu (Tại sao)

- Xác định và giảm mức độ rủi ro của các tính năng chịu lỗi của phần mềm gây ra hiệu suất không thể chấp nhận được.
- Tạo mô tả và kế hoạch phát triển cho một bộ tính năng chịu lỗi có rủi ro thấp.

Sản phẩm và mốc thời gian (Cái gì và khi nào)

- Đến tuần 3:
 - Đánh giá các tùy chọn chịu lỗi
 - Đánh giá các thành phần có thể tái sử dụng

- Bản nháp mô tả đặc điểm khối lượng công việc
- Kế hoạch đánh giá cho việc tập luyện nguyên mẫu
- Mô tả của nguyên mẫu
- Đến tuần 7:
 - Nguyên mẫu hoạt động với các tính năng chịu lỗi chính
 - Mô phỏng khối lượng công việc
 - Năng lực đo lường và giảm dữ liệu
 - Bản nháp mô tả, kế hoạch cho các tính năng chịu lỗi
- Đến tuần 10
 - Đánh giá và lặp đi lặp lại nguyên mẫu
 - Mô tả, kế hoạch sửa đổi cho các tính năng chịu lỗi

Trách nhiệm (Ai và ở đâu)

- Kỹ sư hệ thống: G. Smith
 - Nhiệm vụ 1, 3, 4, 9, 11, hỗ trợ nhiệm vụ 5, 10
- Lập trình viên dẫn đầu: C. Lee
 - Nhiệm vụ 5, 6, 7, 10 hỗ trợ nhiệm vụ 1, 3
- Lập trình viên: J. Wilson
 - Nhiệm vụ 2, 8, hỗ trợ nhiệm vụ 5, 6, 7, 10

Phương pháp (Làm thế nào)

- Nỗ lực tạo mẫu thiết kế theo lịch trình
- Được thúc đẩy bởi các giả thuyết về tác động của khả năng chịu lỗi đối với hiệu suất
- Sử dụng hệ điều hành thời gian thực, thêm các tính năng chịu lỗi nguyên mẫu
- Đánh giá hiệu suất so với khối lượng công việc đại diện
- Tinh chỉnh nguyên mẫu dựa trên các kết quả quan sát

Tài nguyên (Bao nhiêu)

- \$60K - Kỹ sư hệ thống toàn thời gian, lập trình viên dẫn đầu, lập trình viên (10 tuần) * (3 nhân viên) * (\$2K / nhân viên-tuần)
- \$0K - 3 máy trạm chuyên dụng (từ nhóm dự án)
- \$0K - 2 bộ xử lý mục tiêu (từ nhóm dự án)
- \$0K - 1 bộ tiền xử lý thử nghiệm (từ nhóm dự án)
- \$10K - Điều kiện bất khả kháng
- \$70K - Tổng cộng

V. Theo dõi và Giám sát Rủi ro Dự án