

triển.

- Phương pháp phát triển phần mềm nhanh đảm bảo rằng chất lượng của sự phát triển được duy trì
- Quá trình này hoàn toàn dựa trên tiến trình gia tăng. Vì vậy, khách hàng và nhóm biết chính xác những gì được hoàn thành và những gì không. Điều này làm giảm rủi ro trong quá trình phát triển.

5. Hạn chế của mô hình thác nước:

- Nó không phải là một mô hình lý tưởng cho một dự án kích thước lớn
- Nếu yêu cầu không rõ ràng ngay từ đầu thì đó là phương pháp kém hiệu quả hơn.
- Rất khó di chuyển trở lại cái giai đoạn trước đó để thay đổi .
- Quá trình thử nghiệm bắt đầu khi quá trình phát triển kết thúc. Do đó, nó có nguy cơ cao của các lỗi được tìm thấy sau giai đoạn phát triển, và rất tốn kém để sửa các lỗi.

6. Hạn chế của mô hình Agile

- Nó không phải là phương pháp hữu ích cho các dự án phát triển nhỏ.
- Nó đòi hỏi một chuyên gia để có những quyết định quan trọng trong cuộc họp.
- Chi phí thực hiện một phương pháp nhanh hơn một chút so với các phương pháp phát triển khác.
- Dự án có thể dễ dàng đi theo chiều hướng xấu nếu người quản lý dự án không rõ ràng kết quả họ muốn.

7. Sự khác biệt giữa mô hình Agile và Waterfall:

Agile	Waterfall
Nó tách vòng đời phát triển dự án thành chạy nước rút.	Quá trình phát triển phần mềm được chia thành các giai đoạn riêng biệt.
Nó theo một cách tiếp cận gia tăng	Phương pháp thác nước là một quá trình thiết kế tuần tự.
Phương pháp nhanh được biết đến với tính linh hoạt của nó.	Thác là một phương pháp phát triển phần mềm có cấu trúc nên hầu hết thời gian nó có thể khá cứng nhắc.
Agile có thể được coi là một bộ sưu tập của nhiều dự án khác nhau.	Phát triển phần mềm sẽ được hoàn thành như một dự án duy nhất.
Agile là một phương pháp khá linh hoạt cho phép thay đổi được thực hiện trong các yêu cầu phát triển dự án ngay cả khi kế hoạch ban đầu đã được hoàn thành.	Không có phạm vi thay đổi các yêu cầu khi phát triển dự án bắt đầu.
Phương pháp nhanh , theo một cách tiếp cận phát triển lặp lại vì quy hoạch, phát triển, tạo mẫu và các giai đoạn phát triển	Tất cả các giai đoạn phát triển dự án như thiết kế, phát triển, thử nghiệm, vv được hoàn



Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.



↑ +2 ↓



Agile	Waterfall
Phát triển nhanh là một quá trình trong đó các yêu cầu được dự kiến sẽ thay đổi và phát triển.	Phương pháp này là lý tưởng cho các dự án có yêu cầu nhất định và thay đổi không được mong đợi.
Trong phương pháp Agile, thử nghiệm được thực hiện đồng thời với phát triển phần mềm.	Trong phương pháp này, giai đoạn "Thử nghiệm" xuất hiện sau giai đoạn "Xây dựng"
Agile giới thiệu tư duy sản phẩm, nơi sản phẩm phần mềm đáp ứng nhu cầu của khách hàng cuối cùng và thay đổi chính nó theo nhu cầu của khách hàng.	Mô hình này cho thấy một tư duy dự án và đặt trọng tâm của nó hoàn toàn vào việc hoàn thành dự án.
Agile methodology hoạt động đặc biệt tốt với Time & Materials hoặc tài trợ không cố định. Nó có thể làm tăng căng thẳng trong các kịch bản giá cố định.	Giảm rủi ro trong các hợp đồng giá cố định của công ty bằng cách nhận được thỏa thuận rủi ro vào đầu quá trình.
Thích các nhóm nhỏ nhưng chuyên dụng với mức độ phối hợp và đồng bộ hóa cao.	Phối hợp / đồng bộ hóa nhóm rất hạn chế.
Chủ sở hữu sản phẩm với nhóm chuẩn bị các yêu cầu chỉ là về mỗi ngày trong một dự án.	Phân tích kinh doanh chuẩn bị các yêu cầu trước khi bắt đầu dự án.
Đội kiểm tra có thể tham gia vào các yêu cầu thay đổi mà không có vấn đề gì.	Thật khó để thử nghiệm bắt đầu bất kỳ thay đổi nào về yêu cầu.
Mô tả chi tiết dự án có thể được thay đổi bất cứ lúc nào trong quá trình SDLC.	Mô tả chi tiết cần thực hiện phương pháp tiếp cận phát triển phần mềm thác nước.
Các thành viên của Nhóm Agile có thể hoán đổi cho nhau, do đó, chúng hoạt động nhanh hơn. Cũng không cần thiết cho các nhà quản lý dự án vì các dự án được quản lý bởi toàn bộ nhóm	Trong phương pháp thác nước, quy trình luôn đơn giản như vậy, người quản lý dự án đóng một vai trò thiết yếu trong mọi giai đoạn của SDLC.

8. Phần kết luận:

Agile và Waterfall là các phương pháp phát triển phần mềm rất khác nhau và tốt theo cách tương ứng.

Tuy nhiên, có một số khác biệt lớn được đánh dấu bên dưới

- Mô hình thác nước lý tưởng cho các dự án đã xác định được yêu cầu và không có thay đổi nào trong quá trình phát triển. Mặt khác, Agile là phù hợp nhất, nơi có nhiều cơ hội thay đổi yêu cầu thường xuyên hơn.
- Thác nước dễ quản lý, tuần tự và cứng nhắc.



Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.



↑ +2 ↓



thể thực hiện được trong phương thức Waterfall.

All rights reserved



Bài viết liên quan

[MÔ HÌNH XOẺ ỐC LÀ GÌ? KHI NÀO NÊN SỬ DỤNG? Ứ...](#)

[Ngọc Hà](#)

7 phút đọc

👁 17267 📄 0 💬 0 ⬇ 2

[Các mô hình phát triển phần mềm](#)

[HanSora](#)

23 phút đọc

👁 73722 📄 33 💬 8 ⬇ 50

[Tìm hiểu về Mô hình thác nước trong phát triển phần mềm \(STLC\)](#)

[Le Thi Thuy](#)

4 phút đọc

👁 28987 📄 4 💬 0 ⬇ 1

[Scrum và quy trình phát triển phần mềm truyền thống \(SDLC\)](#)

[Nguyễn Thị Phương Linh](#)

13 phút đọc

👁 7482 📄 4 💬 4 ⬇ 4

[Thuận lợi và khó khăn khi sử dụng một số mô hình phát triển phần mềm](#)

[Phạm Thanh Tra](#)

10 phút đọc

👁 2024 📄 3 💬 0 ⬇ 2

Bài viết khác từ Nguyễn Thị Tuyen

[Một vài trường hợp chung để Kiểm thử video](#)

[Nguyễn Thị Tuyen](#)

5 phút đọc

👁 504 📄 1 💬 0 ⬇ 1

[Hướng dẫn kiểm tra UI trên điện thoại di động](#)

[Nguyễn Thị Tuyen](#)

22 phút đọc

👁 1271 📄 1 💬 0 ⬇ 1

[Kiểm tra Cơ sở dữ liệu \(Dữ liệu\) với Các Mẫu Thử...](#)

[Nguyễn Thị Tuyen](#)

15 phút đọc

👁 171 📄 0 💬 0 ⬇ 1

[Kiểm tra Cơ sở dữ liệu \(Dữ liệu\) với Các Mẫu Thử...](#)

[Nguyễn Thị Tuyen](#)

12 phút đọc

👁 506 📄 0 💬 0 ⬇ 0

[TOP 50 Phần mềm test hữu ích cho Tester/QA - Phần 4](#)

[Nguyễn Thị Tuyen](#)

9 phút đọc

👁 350 📄 0 💬 0 ⬇ 0

[TOP 50 Phần mềm test hữu ích cho Tester/QA - Phần 3](#)

[Nguyễn Thị Tuyen](#)

9 phút đọc

👁 267 📄 0 💬 0 ⬇ 0

[TOP 50 Phần mềm test hữu ích cho Tester/QA - Phần 2](#)

[Nguyễn Thị Tuyen](#)

10 phút đọc

👁 1487 📄 0 💬 0 ⬇ 2

[TOP 50 Phần mềm test hữu ích cho Tester/QA - Phần 1](#)

[Nguyễn Thị Tuyen](#)

11 phút đọc

👁 1574 📄 0 💬 0 ⬇ 1



Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.



↑ +2 ↓



TÀI NGUYÊN

[Bài viết](#)[Câu hỏi](#)[Videos](#)[Thảo luận](#)[Công cụ](#)[Trạng thái hệ thống](#)[Tổ chức](#)[Tags](#)[Tác giả](#)[Đề xuất hệ thống](#)[Machine Learning](#)

DỊCH VỤ

[Viblo Code](#)[Viblo CV](#)[Viblo CTF](#)[Viblo Learning](#)

ỨNG DỤNG DI ĐỘNG



LIÊN KẾT



Tiếng Việt

[Về chúng tôi](#)[Phản hồi](#)[Giúp đỡ](#)[FAQs](#)[RSS](#)[Điều khoản](#)[DMCA PROTECTED](#)

© Viblo 2022



Hãy đăng ký một tài khoản Viblo để nhận được nhiều bài viết thú vị hơn.



↑ +2 ↓

