**PHẦN I: SILDE TRÊN LỚP**

## Khái niệm mô hình phát triển tuần tự(Thác nước, chữ V)

* **Mô hình phát triển tuần tự** :
* Mô hình được phát triển theo tuyến của nhiều giai đoạn rời rạc và kế tiếp nhau, mà không có sự chồng chéo.
* Có 2 loại mô hình tiêu biểu đại diện cho mô hình phát triển tuần tự : Mô hình thác nước , Mô hình chữ V.

## Mô hình thác nước là gì?

* **Khái niệm**
* Mô hình thác nước là một mô hình có tính trình tự trong phát triển phần mềm. Nó chú trọng vào sự phát triển logic và lần lượt trong suốt vòng đời phát triển phần mềm (software development life cycle-SDLC), trong đó những pha (phase) ở phía trước phải được hoàn thành trước khi những pha tiếp theo có thể bắt đầu. Mỗi pha trong mô hình thác nước có vai trò khác nhau trong SDLC.

### Ưu điểm

Mô hình thác nước có một vài ưu điểm như sau:

1. Dễ dàng triển khai và quy trình dễ hiểu
2. Với những dự án nhỏ, mô hình thác nước hoạt động hiệu quả và cho kết quả rất tốt.
3. Vì quá trình phát triển rất chặt chẽ, do đó chất lượng mỗi phần và cả dự án sẽ rất chặt chẽ.
4. Yêu cầu đầu vào và đầu ra có sự rõ ràng, nên tiến trình làm việc rất dễ dàng và chất lượng
5. Kết quả được ghi chép dễ dàng

### Nhược điểm

Tuy nhiên, mô hình thác nước cũng còn 1 vài nhược điểm sau:

1. Do pha đầu tiên là phân tích yêu cầu, nên sẽ **không thể thay đổi yêu cầu** ở các pha tiếp theo trong mô hình
2. **Mô hình thác nước không phù hợp để triển khai với những dự án dài và phức tạp**, có nhiều sự thay đổi trong yêu cầu
3. **Khó di chuyển qua lại giữa các pha**. Ví dụ, nếu phần mềm chuẩn bị bước vào pha kiểm thử và có sự thay đổi ở yêu cầu thì sẽ rất khó để quay lại và thay đổi. Việc quay lại sẽ ngốn một khoảng thời gian và công sức rất lớn của team phát triển
4. **Rủi ro cao hơn.** Do quá trình kiểm thử trong mô hình thác nước xuất hiện gần cuối, các chiến thuật để giảm thiểu rủi ro không được triển khai từ sớm. Trong khi ở những mô hình linh hoạt, quá trình kiểm thử diễn ra song song với quá trình phát triển do đó việc giảm thiểu rủi ro hiệu quả hơn rất nhiều

## Mô hình chữ V là gì?

* **Khái niệm**
* V model là sự mở rộng của mô hình thác nước. Không giống như mô hình thác nước. Ở V model, tương ứng với một giai đoạn kiểm thử là một giai đoạn phát triển phần mềm, thử nghiệm trong mô hình chữ V được thực hiện song song với chu kì phát triển phần mềm.
* Ưu điểm.  
  – Đơn giản dễ sử dụng.  
  – Có hoạt động, kế hoạch cụ thể cho quá trình test.  
  – Tiết kiệm được thời gian, và có cơ hội thành công cao hơn waterfall.  
  – Chủ động trong việc phát hiện bug, sớm tìm ra bug ngay từ những bước đầu.
* Nhược điểm.  
  – Độ linh hoạt ít, còn tồn tại sự cứng nhắc. Nó thể hiện ở chỗ cứ sau mỗi step thì lại phải có một  công đoạn test.

Nếu yêu cầu dự án không quá phức tạp và dễ hiểu, thì việc thực hiện nhiều công đoạn test như vậy là tốn thời gian.  
– Giống với water fall, sản phẩm của dự án chỉ được xuất hiện khi tất cả các bước được hoàn thành xong, không có nguyên mẫu ngay từ ban đầu.

Không đáp ứng được yêu cầu dịch vụ vừa phát triển song song với vừa bán sản phẩm.  
– Nếu có sự thay đổi về kỹ thuật ở nửa chừng, thì sẽ phải quay lại các bước đầu tiên, thực hiện lại, update lại tài liệu.

* **Mô hình phát triển lặp lại**
* **Khái niệm**
* Là 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.

### 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.
* **Mô hình phát triển gia tăng**
* **Khái niệm**
* 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.
* **Ưu điểm**

– Có thể sớm tạo ra nguyên mẫu của sản phẩm trong vòng đời phát triển của nó.  
– Độ linh hoạt cao hơn và khi thay đổi yêu cầu dự án thì chi phí sẽ ít hơn nhiều, vì những thay đổi thuộc về module nào thì module đó sẽ thay đổi mà các module khác không hề bị ảnh hưởng.  
– Việc phân chia thành các module cũng sẽ làm cho việc test nhẹ nhàng hơn, những module đơn giản thì test cũng đơn giản, sớm kết thúc.  
– Giảm chi phí cho lần đầu giao sản phẩm.  
– Dễ dàng quản lý các rủi ro có thể phát sinh.

* **Nhược điểm**

– Cần phải có những khả năng thiết kế tốt và phương pháp tốt, để có thể hiểu rõ được yêu cầu và biết cách phân chi nó ra như thế nào cho hợp lý.  
– Chi phí để phát triển theo phương pháp này là rất cao, cao hơn hẳn waterfall.

* **Mô hình phát triển Scrum**
* **Khái niệm**
* Scrum **là một phương pháp phát triển phần mềm linh hoạt, được dựa theo cơ chế lặp và tăng trưởng**. Scrum được thiết kế để hỗ trợ việc tạo ra, phân phối và cải tiến các sản phẩm phức tạp. Bản thân mô hình Scrum cũng là một Framework – khuôn khổ cơ bản.
* **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.

Tiếp Theo

1. **SDLC LÀ GÌ?**

* SDLC là viết tắt của Softwware Development Life Cycle( vòng đời phát triển phần mềm). SDCL là một quy trình phát triển phần mềm với chất lượng cao và chi phí thấp nhất trong thời gian ngắn nhất có thể.
* SDLC là chia nhỏ quy trình phát triển phần mềm thành các giai đoạn, từ ý tưởng đến triển khai. SDLC được sử dụng để giúp các nhóm phát triển tối ưu hóa quy trình làm việc của họ và làm cho có thể mở rộng mà vẫn đảm bảo chất lượng.
* SDLC gồm 7 giai đoạn chính:
* Thu thập và phân tích thông tin ( Requirement Analysis)
* Lập kế hoạch (Planning hay Defining)
* Thiết kế (Designing)
* Phát triển(Development)
* Kiểm thử (Testing)
* Triển khai (Deployment)
* Bảo trì
* Các phương pháp SDLC
  + Mô hình thác nước (Waterfall)
  + Phương pháp Agile
  + Phương pháp lặp lại (Iterative)
  + Phương pháp hình chữ V (V-shaped)
  + Phương pháp xoắn ốc (spiral)…

1. **Nêu tên các mô hình đã được học?**

* Mô hình phát triển Agile
* Mô hình phát triển Scrum

1. **Giải thích các giai đoạn của mô hình thác nước.**

Diagram, text

Description automatically generated with medium confidence

**System engineering( Xác định yêu cầu):** Thu thập và phân tích yêu cầu được ghi lại vào tài liệu đặc tả yêu cầu trong giai đoạn này nhằm phục vụ cho quá trình phát triển trong tương lai.

Tài liệu đặc tả yêu cầu xác định ứng dụng sẽ hoạt động những gì, nhưng không cụ thể nó sẽ hoạt động như thế nào.

**System Analysis( Phân tích và Thiết kế):**

* Phân tích: Hệ thống được phân tích để có thể tạo ra một mô hình và logic của hệ thống phù hợp mà sẽ được sử dụng trong ứng dụng.
* Thiết kế: Giải quyết các yêu cầu thiết kế về kỹ thuật, như ngôn ngữ lập trình, tầng dữ liệu, dịch vụ,…Một bản thiết kế điển hình sẽ được hoàn thành một cách càng cụ thể càng tốt. Nó sẽ mô tả chính xác và logic của hệ thóng được đề cập trong phần phân tích sẽ được thực thi như nào.

**Coding(Viết mã):** 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(Kiểm thử sản phẩm):** Cài đặt và kiểm thử phần mềm. Công việc chính của giai đoạn này là kiểm tra và sửa tấy cả những lỗi tìm được sao cho phần mềm họa động chính xác và đúng theo tài liệu đặc tả yêu cầu.

**Implementation(Triển khai ứng dụng):** Triển khai hệ thống trong môi trường khách hàng và đưa ra thị trường.

**Operations and Maintenance(Bảo trì hệ thống):** Bảo trì hệ thống khi có bất kỳ thay đổi nào từ phía khách hàng, người sử dụng.

1. Nêu các giai đoạn phát triển trong mô hình phát triền Scrum

Diagram

Description automatically generated

**Phân tích mô hình phát triển Scrum**

* Chia các yêu cầu ra làm theo từng giai đoạn. Mỗi 1 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( Không 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.

**Qui trình: Quy định cách thức vận hành của SCRUM.**

* Sprint Planning meeting: Hoạt định cho mỗi giai đoạn.
* Review: Tổng kết cho mỗi giai đoạn.
* Daily Scrum Meeting: Review hàng ngày.

1. Nêu cách tổ chức trong mô hình phát triển Scrum

Tổ chức dự án

* Product Owner(PO)
* 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.
* Scrum Master là người đảm bảo các quy trình của Scrum được thực hiện đúng và thuận lợi.
* Development Team(Dev Team)
* Một nhóm từ 4-7 kỹ sư phầm 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
* 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.

Tiếp Theo

* **Giới thiệu bảng Kanban**
* **Khái niệm**
  + Kanban là một phương pháp Agile nhưng không nhất thiết cần có tính lặp. Các quy trình như Scrum có các lần lặp ngắn (Sprint) là vòng đời của dự án trên quy mô nhỏ, có điểm bắt đầu và kết thúc riêng biệt cho mỗi lần lặp. Kanban cho phép phần mềm được phát triển trong một chu kỳ phát triển lớn.
  + Kanban là một ví dụ về một phương pháp Agile vì nó đáp ứng tất cả mười hai nguyên tắc đằng sau tuyên ngôn Agile, bởi vì mặc dù nó không có tính lặp, nhưng vẫn có tính tăng trưởng.
  + Khác với mô hình Scrum là giới hạn thời gian cho phép để hoàn thành một công việc cụ thể (sprint) thì Kanban giới hạn số lượng công việc cho phép trong một điều kiện nhất định (bao gồm nhiều task trên một thẻ Kanban).
* **Ưu/ Nhược điểm của bảng Kanban**
  + **Ưu Điểm** :
    - Mọi người đều ở trên cùng một mặt phẳng: Tất cả các nhiệm vụ đều dễ dàng nhìn thấy, điều này mang lại sự minh bạch cho toàn bộ quá trình làm việc. Mỗi thành viên có thể cập nhật nhanh về trạng thái của mọi dự án hoặc nhiệm vụ.
    - Chu kì thời gian làm việc được rút ngắn lại: Trong Kanban, không phải mỗi người nắm giữ một kỹ năng, vì như vậy nếu người đó không hoàn thành tốt công việc thì sẽ sở thành điểm tắc nghẽn trong quy trình làm việc. Vì vậy, nhóm Kanban luôn hỗ trợ và bổ sung kỹ năng cho nhau, đảm bảo các thành viên luôn được học hỏi và không chỉ tập trung vào kỹ năng nào.
    - Ít tắc nghẽn hơn: Kanban là giới hạn số lượng công việc đang thực hiện.Giới hạn công việc đang tiến hành giúp ít tắc nghẽn và tăng dự phòng trong quy trình của nhóm do thiếu tập trung, con người hoặc kỹ năng.
    - Hệ thống đơn giản, dễ hiểu: Kanban đặc biệt linh hoạt ở chỗ có thể dễ dàng sử dụng ở bất kỳ đội nhóm trong ngành nghề hay quy mô nào vì nó dễ sử dụng.
    - Hệ thống phản ứng nhanh nhạy: Kanban giúp chúng ta dễ dàng đáp ứng các yêu cầu thay đổi liên tục của khách hàng. Nó cho phép thay đổi các ưu tiên, tổ chức lại hoặc chuyển nhiệm vụ trọng tâm thực sự nhanh chóng.
    - Yêu cầu tập trung vào các nhiệm vụ hiện tại cho đến khi hoàn thành: Điều này nhờ vào khái niệm giới hạn công việc đang xử lý. Giới hạn công việc đang xử lý giúp thúc đẩy các nhóm cộng tác để hoàn thành các mục công việc nhanh hơn, mặt khác giúp loại bỏ các phiền nhiễu như đa nhiệm.
  + **Nhược điểm**:
    - Kanban thường tập trung vào các tác vụ hàng ngày, nó thúc đẩy việc thực thi nhiệm vụ, nhưng có thể gây rủi ro cho bức tranh lớn hơn là chiến lược và những kết quả quan trọng nhất. Các nhiệm vụ nhỏ được hoàn thành, nhưng kết quả cuối cùng chưa chắc đã đạt được.
    - Không có khung thời gian của từng giai đoạn: Vì các cột chỉ được gắn nhãn với các giai đoạn (phải thực hiện, đang thực hiện, hoàn thành), nên có thể khó thấy khi nào mọi việc sẽ được thực hiện.
    - Phải cập nhật bảng: Các nhóm phải nhấn mạnh tầm quan trọng của việc cập nhật bảng, nếu không, họ có nguy cơ làm việc với thông tin không chính xác.
    - Kanban sẽ trở nên rất khó áp dụng vì có quá nhiều hoạt động hoặc nhiệm vụ liên quan đến nhau trong một hệ thống. Việc tăng cường chuyển giao tính năng và chuyên môn giữa các nhiệm vụ khác nhau quá thường xuyên và làm tăng khó khăn để theo kịp tất cả các hoạt động.
    - Đầu ra có thể không đảm bảo chất lượng: Kanban hoạt động giống như một cấu trúc giám sát giúp cho các luồng công việc trôi chảy hơn. Nếu bất kỳ công việc nào được thực hiện là không thỏa đáng, yêu cầu làm lại có thể làm tình hình tồi tệ hơn vì đòi hỏi nhiều thời gian và nguồn lực hơn để hoàn thành.
* **Khái niệm mô hình phát triển Agile**
* **Khái niệm**
  + Agile (Agile Software Development) có nghĩa là phương thức phát triển phần mềm linh hoạt, được ứng dụng trong quy trình phát triển phần mềm với mục tiêu là đưa sản phẩm đến tay người dùng càng nhanh càng tốt.
  + Agile giống như một phương pháp luận, một triết lý dựa trên hơn nguyên tắc phân đoạn vòng lặp (iterative) và tăng trưởng (incremental).
  + Triết lí Agile đã vượt xa khỏi khu vực truyền thống của mình là phát triển phần mềm để đóng góp sự thay đổi trong cách thức làm việc, quản lý, sản xuất ở các ngành khác như sản xuất, dịch vụ, sales, marketing, giáo dục... và trở thành một phương thức quản lý dự án phổ biến nhất hiện nay với nhiều đại diện được gọi là các phương pháp “họ Agile”
* **Ưu điểm:**
* **Thực hiện thay đổi dễ dàng**: Bởi vì dự án được chia thành các 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 rõ ràng, vì việc này không 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 việc 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 các ý kiến và phản hồi, từ đó họ sẽ có ảnh hưởng một 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 hàng 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 lần nếu cần.
* **Nhược điểm:**
  + **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.
* **Mối liên quan giữa các hoạt động kiểm thử và phát triển trong quy trình phát triển phần mềm**
  + **Vòng đời phát triển phần mềm**:Đây là chuỗi các hoạt động được các Developers thực hiện để thiết kế và phát triển phần mềm có chất lượng tốt. Vòng đời phát triển phần mềm sử dụng thuật ngữ ‘Development’, không chỉ liên quan đến các nhiệm vụ coding được thực hiện bởi các Developers mà còn kết hợp các nhiệm vụ được đóng góp bởi Tester và các bên liên quan. Trong Vòng đời phát triển phần mềm, các test cases được tạo ra bởi các tester.
  + **Vòng đời Kiểm thử phần mềm**: Bao gồm một loạt các hoạt động được thực hiện bởi tester, sử dụng các phương pháp để kiểm thử sản phẩm phần mềm. Vòng đời kiểm thử Phần mềm sử dụng thuật ngữ “Testing”, không chỉ liên quan đến tester, trong một số trường hợp cũng liên quan đến các Developers.

Tiếp Theo

* **Bảng Kanban bắt buộc ít nhất có mấy cột và nếu ý nghĩa của những cột đó là gì ?**
* Nói một cách đơn giản, là một bảng quản lý toàn bộ quy trình làm việc của bạn. Một bảng Kanban cá nhân bao gồm 3 cột rõ ràng:

**To Do (Cần làm), Doing (Đang làm)**và **Done (Đã làm).**Mỗi cột như vậy lại bao gồm nhiều công việc được viết trên một thẻ và được xếp theo tình trạng công việc đó.

## Product owner là gì?

* Product Owner (gọi tắt là PO) như tên tiếng Việt được hiểu là người “sở hữu” sản phẩm, tức là chịu trách nhiệm giải quyết những vấn đề thực tế của người dùng (user) khi sử dụng sản phẩm đó. Ngoài ra, PO vận hành, cải tiến sản phẩm và tối ưu hóa lợi nhuận trên sản phẩm để đạt được mục tiêu của công ty.
* **Scrum Team bao gồm những thành viên có vai trò gì ?**
* Nhóm Scrum gồm 3 vai trò là Product Owner, Development Team và Scrum Master :

1. Product Owner :

* Product Owner có vai trò chịu trách nhiệm **định hướng sản phẩm** trong suốt quá trình sản xuất. Nhiệm vụ của Product Owner là **tối ưu hoá giá trị của sản phẩm** thông qua việc tận dụng tốt nhất khả năng sản xuất của Development Team. Có thể Product Owner chưa cần biết ngay từ đầu thật cụ thể là sẽ làm những gì, nhưng có hiểu biết sâu sắc tại sao lại xây dựng sản phẩm này.
* Product Owner là người đại diện **duy nhất** và **chịu trách nhiệm về sản phẩm đang xây dựng**. Cụ thể, Product Owner là người duy nhất chịu trách nhiệm quản lý Product Backlog – nơi lưu trữ các hạng mục cần phát triển của sản phẩm và tích cực tham gia vào các cuộc họp của nhóm để cung cấp các thông tin cần thiết, chủ động quản lý một Kế hoạch Phát hành cho sản phẩm.

1. Scrum Master :

* Scrum Master là một vai trò then chốt trong nhóm Scrum có trách nhiệm **đảm bảo** Scrum được vận hành đúng, tuân thủ nguyên tắc, các kỹ thuật và quy tắc Scrum nhằm hướng đến kết quả tốt nhất.
* Scrum Master là một người duy nhất. Scrum Master**không trực tiếp** tham gia vào công việc làm ra sản phẩm, nhưng là **kết dính**để các bên phối hợp với nhau tạo ra sản phẩm tốt. Scrum  Master không phải là quản lý của Nhóm mà là một **lãnh đạo theo phong cách phục vụ**. Với vai trò là lãnh đạo phục vụ, Scrum Master cung cấp các dịch vị cho Product Owner, Nhóm Phát triển và tổ chức.

1. Development Team :

* Development Team là đội ngũ **trực tiếp làm ra sản phẩm**, bao gồm các chuyên gia có nhiệm vụ chuyển giao phần tăng trưởng có thể chuyển giao được ở cuối mỗi Sprint.
* Development Team vận hành tốt nhất với số lượng thành viên vừa phải, quy định của Scrum là từ 3 đến 9 người. Development Team trong Scrum cũng là một nhóm tự tổ chức liên chức năng. Team được trao quyền để tự định hướng và đưa ra các quyết định liên quan đến công việc sản xuất. Tự tổ chức cũng có nghĩa là nhóm có toàn quyền lựa chọn công cụ, kỹ thuật và cách thức để hoàn thành công việc. Trong quá trình sản xuất, họ tự tiến hành ước lượng, phân bổ, theo dõi, điều tiết công việc theo hình thức tập thể.
* **1 Sprint có bao nhiêu giai đoạn phát triển phần mềm ?**
* SDLC gồm 7 giai đoạn chính:
* Thu thập và phân tích thông tin ( Requirement Analysis)
* Lập kế hoạch (Planning hay Defining)
* Thiết kế (Designing)
* Phát triển(Development)
* Kiểm thử (Testing)
* Triển khai (Deployment)
* Bảo trì

Tình Huống 1 : Sile 7

Mô Hình Scrum

* **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.

Tình Huống 2 : Sile 8

Mô Hình phát triển lặp

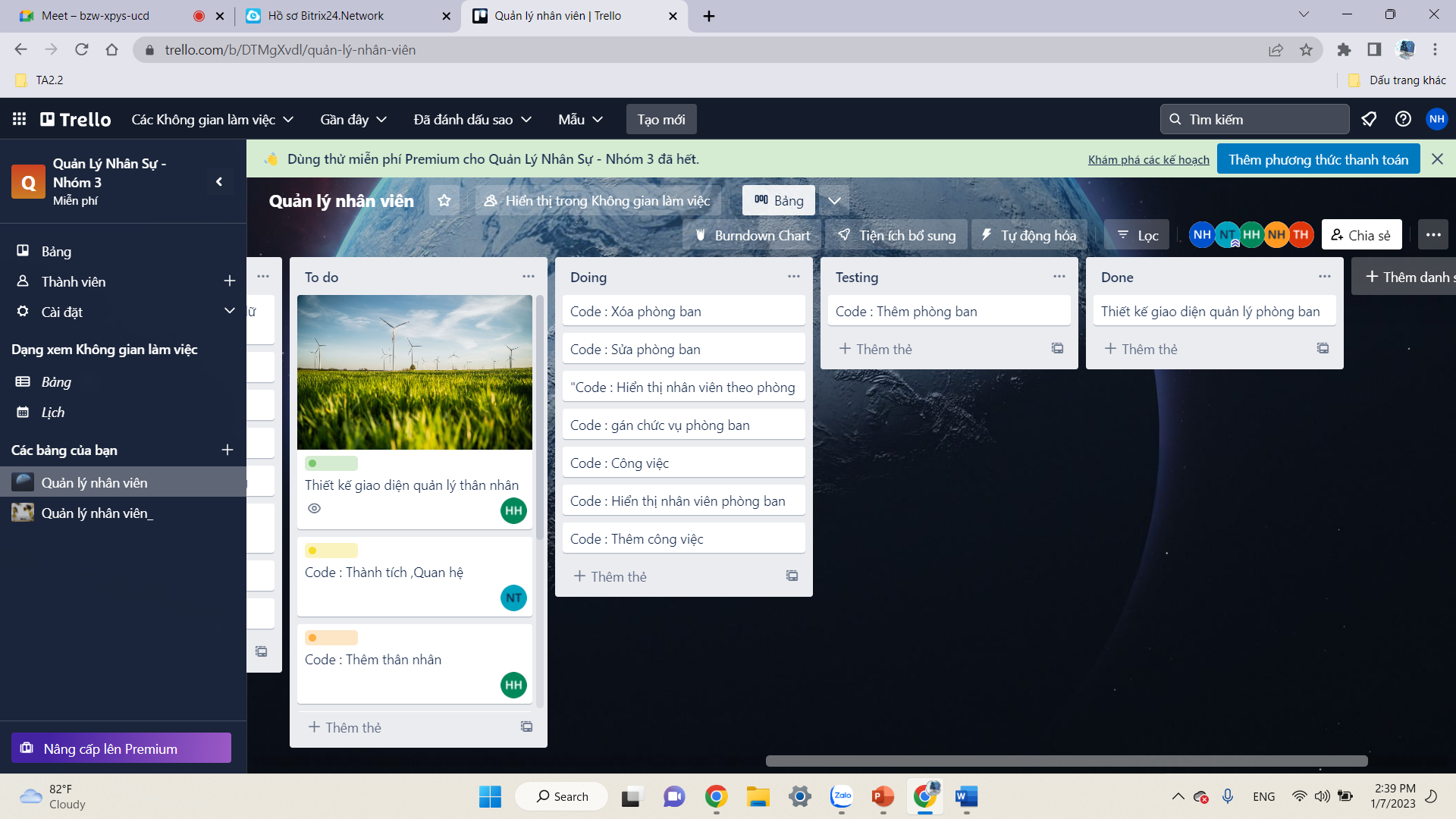
### Ưu điểm

* 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.

Tính Huống 1 sile 13



Tính Huống 2 sile 14

Tính Huống 1 Sile 21

Tính Huống 1 Sile 28

Unit testing bao gồm kiểm thử chức năng :

* như kiểm tra việc tính toán chính xác của thuật toán
* các đặc điểm phi chức năng cụ thể liên quan đến tài nguyên , như rò rỉ bộ nhớ
* kiểm thử hiệu suất , như tính toán hoàn thành đủ nhanh.